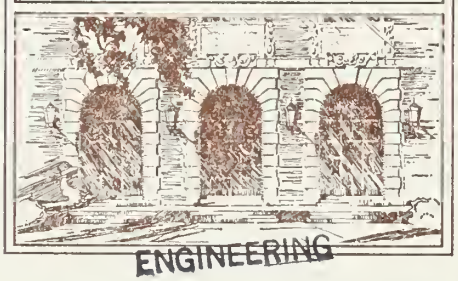


Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

URBANA, ILLINOIS 61801

CAC Document No. 18

PARALLEL RADIATION TRANSPORT CODE A MONTE CARLO METHOD APPLIED TO ILLIAC IV AND ITS STATISTICAL CONSIDERATION

by

Ken'ichi Miura

October 1, 1971

CAC Document No. 18

PARALLEL RADIATION TRANSPORT CODE A MONTE CARLO METHOD APPLIED TO ILLIAC IV AND ITS STATISTICAL CONSIDERATION

By

KEN'ICHI MIURA

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
October 1, 1971

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the U.S. Army Research Office-Durham under contract no. DAHC04 72-C-0001 and was submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, October 1971.

Abstract

A Monte Carlo method to solve radiation transport problems by using the ILLIAC IV is discussed. An emphasis is put on the special features to be encountered in parallel computation: data structure, PE efficiency, etc. A test program implemented in GLYPNIR--an ILLIAC IV language--is shown and some preliminary results on its statistical properties are also discussed.

ACKNOWLEDGEMENTS

I wish to express sincere gratitude to Professor D. L. Slotnick and Mr. D. E. McIntyre for directing this thesis, and also to Professor A. B. Chilton in the Nuclear Engineering Department for useful suggestions and discussions. I especially appreciate the interest and encouragement of David McIntyre during the preparation of this thesis. Additional thanks go to Mr. Duncan Lawrie and Mr. Terry Layman for helping me implement the test program.

Finally, I wish to thank Mrs. Jan Claeys and Mrs. Norma Thompson for an excellent job of typing this thesis and Mr. Fred Hancock for beautiful figures.

This work was supported in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center under Contract No. USAF 30(602)4144.

TABLE OF CONTENTS

Page

1. INTRODUCTION 1

1.1 Basic Equations 2

1.2 The Model for Monte Carlo Calculation (general discussion). 4

1.3 Creation of Particles 5

1.4 Processing a Time Step. 6

1.5 The New Temperature 7

2. RANDOM SAMPLING. 10

2.1 Linear Congruential Method. 10

2.2 Random Sampling from a Given Distribution Function. 12

2.2.1 Rejection Method 12

2.2.2 Method of Inverse Function 14

2.2.3 Approximation of Distribution Function by some Simpler Ones. 14

2.2.4 Root-finding Method. 17

2.3 Some Remarks on Random Sampling 18

2.3.1 Forcing (sampling exactly n samples out of N). . . . 18

2.3.2 Shuffling (or random permutation of data). 19

3. FUNDAMENTAL STRUCTURE OF THE PROGRAM 25

3.1 Basic Program Structures and Their Comparison 25

3.1.1 Geometrical Structure of the Medium and Hardware Structures of the Computer. 25

3.1.2 Particle Migration Scheme. 26

3.1.3 Coordinate Migration Method. 28

Page

3.2 Particle Migration Scheme 28

3.2.1 Particle Tables. 28

3.2.2 Packing Data and Integer-Real Conversion 29

3.2.3 Chain Structure. 29

3.2.4 Material Table 30

3.2.5 I/O. 31

3.2.6 Total Capacity of Particles. 31

4. A TEST PROGRAM FOR ILLIAC IV 42

4.1 Description of Major Subroutines. 42

4.2 Main Program. 45

5. SOME COMPUTATIONAL RESULTS 49

5.1 Determination of Cell Size (ΔX) and Time Interval (Δt) and Their Relation to the Rate of Boundary-Crossing 49

5.2 Effect of Multiple Scattering 50

5.3 An Effect Caused by Modifying a Probability Distribution Function for Survival Distances of Particles . 51

5.4 Skewed Cell Boundaries Vs Straight Cell Boundaries. 51

5.4.1 64 x 8 Cells 52

5.4.2 64 x 64 Cells. 53

5.5 Concluding Remarks. 53

5.5.1 Problem of Table-Overflow and Normalization. 53

5.5.2 Problem of Word Size (alternative format). 54

5.5.3 About the Flexibility of Logical Network Which Interconnects PEs. 54

5.5.4 Variable Contraction Ratio of Cell Sizes 54

Page

LIST OF REFERENCES 65

APPENDICES 67

1. INTRODUCTION

The Monte Carlo Method started its history in 1777 when Conte de Buffon (1707-1788) presented a famous problem to calculate π by tossing needles (Buffon's problem). (Ref. [1]). Real progress was not made until digital computers became available. In 1947 J. von Neumann, R. Richtmyer and S. Ulam used the ENIAC to solve neutron diffusion problems by simulating the movement of particles. (Ref. [12]). Since then, development has been remarkable in fields such as partial differential equations, definite integrals, serving systems, linear equations and transport problems. (Ref. [10]). In transport problems a necessity to calculate neutron flux within reactors or gamma-ray flux through shielding walls has encouraged the development of more efficient algorithms.

Due to their statistical nature, simulation of random processes on a digital computer requires a large memory and fast logical and arithmetic operations to obtain necessary accuracy. The parallel structure of the ILLIAC IV computer, although originally designed for matrix problems or partial differential equations, is applicable to random simulation. This paper is concerned with the application of the ILLIAC IV to radiation transport problems.

In this chapter, a physical model and the basic equations are presented. Certain other assumptions are also discussed. Special features to be encountered in parallel computation will be discussed in the subsequent chapters.

1.1 Basic Equations

The physical phenomenon described in this paper is "non-linear radiation transport" in the low energy region, which is encountered, for example, in the study of stellar atmosphere. (Ref. [2], [4], [7]).

Consider a block of material (or gas, to be referred to as "medium" hereafter) which is not necessarily homogeneous. Let a beam of photons be incident on it (or let a heat source be attached to it). Some of the photons will pass through the medium undisturbed, some will be absorbed by the medium and release their energy to it, while others will be scattered into new directions (with new energies). Since the absorption of photons affects the local temperature and the change of the temperature itself affects the emission of photons, the whole process is rather complicated. Evolution of the photon intensity and the temperature is given in the form of a system of nonlinear integro-differential equations as will be shown below.

Let dE denote the radiation energy in a frequency range $(\nu, \nu, + d\nu)$ which is transported across an element of area dS at x and in directions confined to a solid angle $d\Omega$ during a time dt. (Fig. 1.1). Then the intensity $I(\vec{x},\mu,\nu)$ is defined as follows:

$$dE_\nu = I(\vec{x},\mu,\nu) \cos\Theta \, d\nu dS d\Omega dt$$

The basic equation which I satisfies is:

$$\frac{d}{ds} I(\vec{x},\mu,\nu) = -\sigma_{tot} I(\vec{x},\mu,\nu) + B(\vec{x},\mu,\nu) + \sigma_{scatt} \cdot \iint P(\mu' \to \mu) I(\vec{x},\mu,\nu) d\mu' d\Omega \qquad (1,1)$$

The left hand side of (1,1) is the total change of the intensity along the path of radiation; its more explicit form is dependent on the geometry

which one is to consider (Cartesian, spherical, etc.). Table 1.1 shows some examples.

The first term on the right hand side represents diminution of the radiation caused by various kinds of interaction: absorption, scattering, or pair-creation (only in high energy region). The probability of the occurrence of each interaction is expressed in terms of cross section. σtot is the total cross section and of course $\sigma tot = \sigma abs + \sigma scatt + \ldots$. The second term is the source term which includes the black body radiation (note that the absorption of photons causes re-emission of new photons through this process), pair-annihilation (only in high energy region) and some other heat sources. In astrophysics, the assumption of local thermodynamical equilibrium is often adopted. That is, if the temperature in the medium varies very slowly with relation to position then the laws of thermodynamic equilibrium are still applicable at each point with the local temperature. Therefore, photons are re-emitted according to "Kirchhoff's law:"

$$B(\vec{x},\mu,\nu) = \sigma abs \cdot \frac{2h\nu^3}{c^2} \frac{1}{\exp\left(\frac{h\nu}{kT}\right) - 1}$$

where σabs and T depend on $\vec{x}$.

Note that the temperature T is involved in a non-linear way.

The third term is the scattering term and represents a contribution from other directions μ' into μ. P denotes the scattering probability from μ' to μ. If Thomson scattering is assumed, then

$$P(\cos\Theta) = \frac{3}{16} (1 + \cos^2\Theta)$$

and

$$\mu' = \mu \cdot \cos\Theta - \sqrt{1-\mu'^2} \sin\Theta \cos(\psi'-\psi)$$

(See Fig. 1.2)

The temperature T, on the other hand, changes with time according to the following equation:

$$\rho Cv\frac{\partial T}{\partial t} = \frac{2h}{c^2}\int_0^\infty \sigma abs\ \nu^3 \cdot \frac{(-1)}{\exp\left(\frac{h\nu}{kT}\right) - 1}\ d\nu$$

where

$$+ \int_{-1}^{1} d\nu \int_0^\infty \sigma abs \cdot I(\vec{x},\mu,\nu)\ d\nu \qquad (1,2)$$

ρ : density

Cv : specific heat

σabs : absorption cross section
(dependent on $\bar{x}$ and T also depends on $\vec{x}$)

The left hand side of (1,2) is the change of stored energy in the medium. The first term on the right hand side is the loss of energy due to the black body radiation. The second term is the gain of energy due to the absorbed photons. Hence, (1,1) and (1,2) describe the energy balance of the total system. (Fig. 1.3). If I is averaged over frequency, then those equations become simpler . That is,

$$\frac{d}{ds} I(\vec{x},\mu,t) = -(\sigma abs + \sigma scatt)\ I(\vec{x},\mu,t) + \sigma scatt \int_{-1}^{1} I(\vec{x},\mu',t) P(\mu,\mu')d\mu' + \sigma abs \frac{\sigma stephan}{\pi} T(\vec{x},t)^4 + S(\vec{x},\mu) \qquad (1,1)'$$

and

$$\rho Cv \frac{\partial T(\vec{x},t)}{\partial t} = -\sigma abs \frac{\sigma stephan}{\pi} T(x,t)^4 + \sigma abs \int_{-1}^{1} I(\vec{x},\mu',t)d\mu' \qquad (1,2)'$$

1.2 The Model for Monte Carlo Calculation (general discussion)

This section describes the model implemented on ILLIAC IV.

(1) The geometry is two dimensional. (Density is assumed to be uniform with regard to the third coordinate although particles

travel in three dimensional space).

(2) The material is expressed as 64 x 8 cells (it is easy to extend up to 64 x 64); each cell can have variable density and specific heat.

(3) Absorption is due to the photo-electric effect, and its cross section can be factored as a density dependent term and another term which is dependent on photon energy. The same thing is assumed as for scattering cross section.

$$\sigma abs = \rho(x,y) \cdot \kappa(E)$$

$$\sigma scatt = \rho(x,y) \cdot \lambda(E)$$

$\rho(x,y)$, $\kappa(E)$ and $\lambda(E)$ can be stored as tables and linear interpolation is used to generate cross sections.

(4) The number of particles which are to be created is dependent on the local temperature. There are two alternatives as to their energies: Planckian distribution or mono-energetic (the latter is adopted for the test program).

Photons are modeled as a particle table. The term "particles" should be distinguished from "photons" hereafter. The former means pseudo-photons within a computer whereas the latter means actual photons. The table size is limited by the size of memory available. Refer to Chapter 3 for the actual table size. The medium has been divided into small cells of $\Delta x \cdot \Delta x$. The time unit is Δt.

1.3 Creation of Particles

At the beginning of a time step, the number of particles to be created in cells is determined according to the temperature. As the table size is limited, more particles are assigned to those cells which are hot. Therefore, only the cell numbers are assigned to the new particles at this

stage.

The creation of new particles follows. The quantities given below are assigned to each of the new particles:

(1) relative coordinates x and y . . . . (uniformly distributed)

(2) direction cosine μ . . . . . (uniformly distributed)

(3) sine and cosine of azimuthal angle . (uniformly distributed)

(4) survival distance (path length). . . (exponentially distributed)

(5) fraction ξ to indicate at what time $\xi\Delta t$ this particle was created. . . . . (uniformly distributed)

1.4 Processing a Time Step

After all the new particles are created, the table scanning starts. The position of a particle is incremented as follows:

$$x' = x + v\mu\Delta t$$

$$y' = y + v\sqrt{1-\mu^2}\sin\phi\cdot\Delta t$$

The distance which a particle is supposed to travel ($l = v\Delta t$) is compared with $\tau/\sigma tot$ where $\sigma tot = \sigma abs + \sigma scatt$, and if the former is larger than the latter an interaction is assumed to occur. The type of interaction is determined by generating a uniformly distributed random number and comparing it with $\sigma abs/\sigma tot$. If the random number is less than this quantity, absorption is taken and the energy of the particle is added to the cell in which the interaction occurred. This particle is to be destroyed.

On the other hand, if the random number is greater than this quantity, then the interaction is scattering and new angles, new survival distances are selected and assigned to the particle.

1.5 The New Temperature

After the scanning of the particle table, the intensity in each cell is computed by summing the particles in it. Then the new temperature is computed.

The last three sections are a narrative description of the Monte Carlo program. A flow chart of the computation process (not the program) is shown in Figure 1.4. In the subsequent chapters particular problems for parallel processing are to be discussed.

Table 1.1

$\frac{d}{ds} = \frac{1}{c}\frac{\partial}{\partial t} + \mu\frac{\partial}{\partial x}$	(slab-symmetry)
$\frac{1}{c}\frac{\partial}{\partial t} + \mu\frac{\partial}{\partial r} + \frac{1-\mu^2}{r}\frac{\partial}{\partial \mu}$	(spherical symmetry)
$\frac{1}{c}\frac{\partial}{\partial t} + \sin\phi\left(\mu\frac{\partial}{\partial r} + \frac{1-\mu^2}{r}\right) + \cos\phi\frac{\partial}{\partial z}$	(cylindrical symmetry)

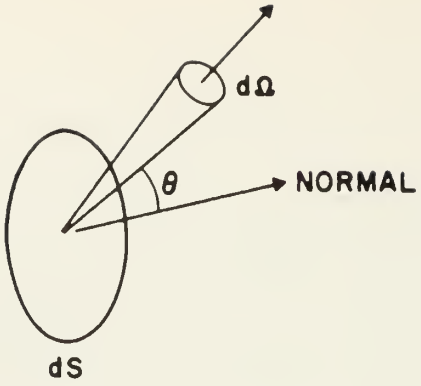


Figure 1.1 Definition of Intensity

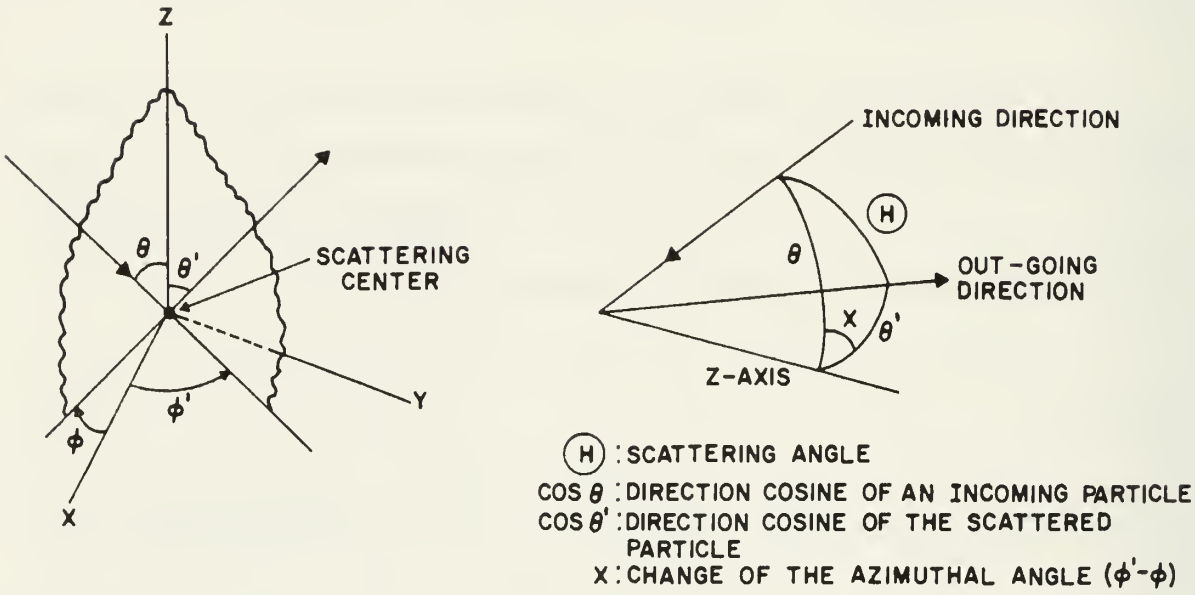


Figure 1.2 Definitions of Angles in Scattering Process

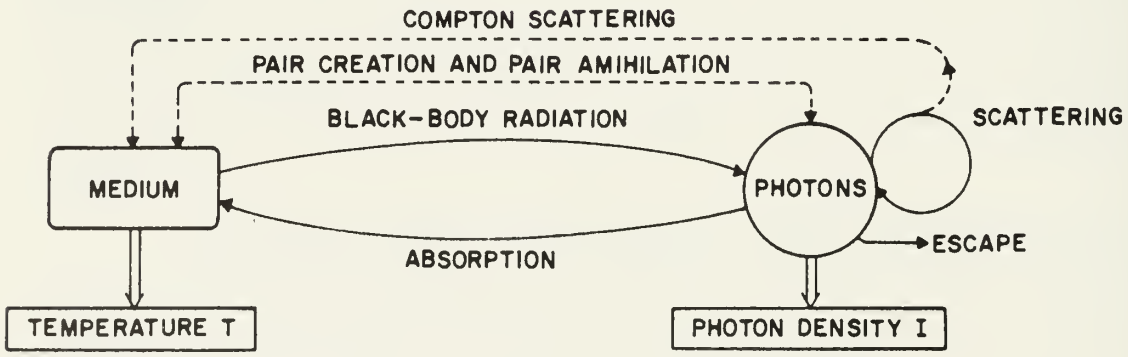


Figure 1.3 Flow Diagram of Conversion of Energy
(Dotted lines are for high energy case)

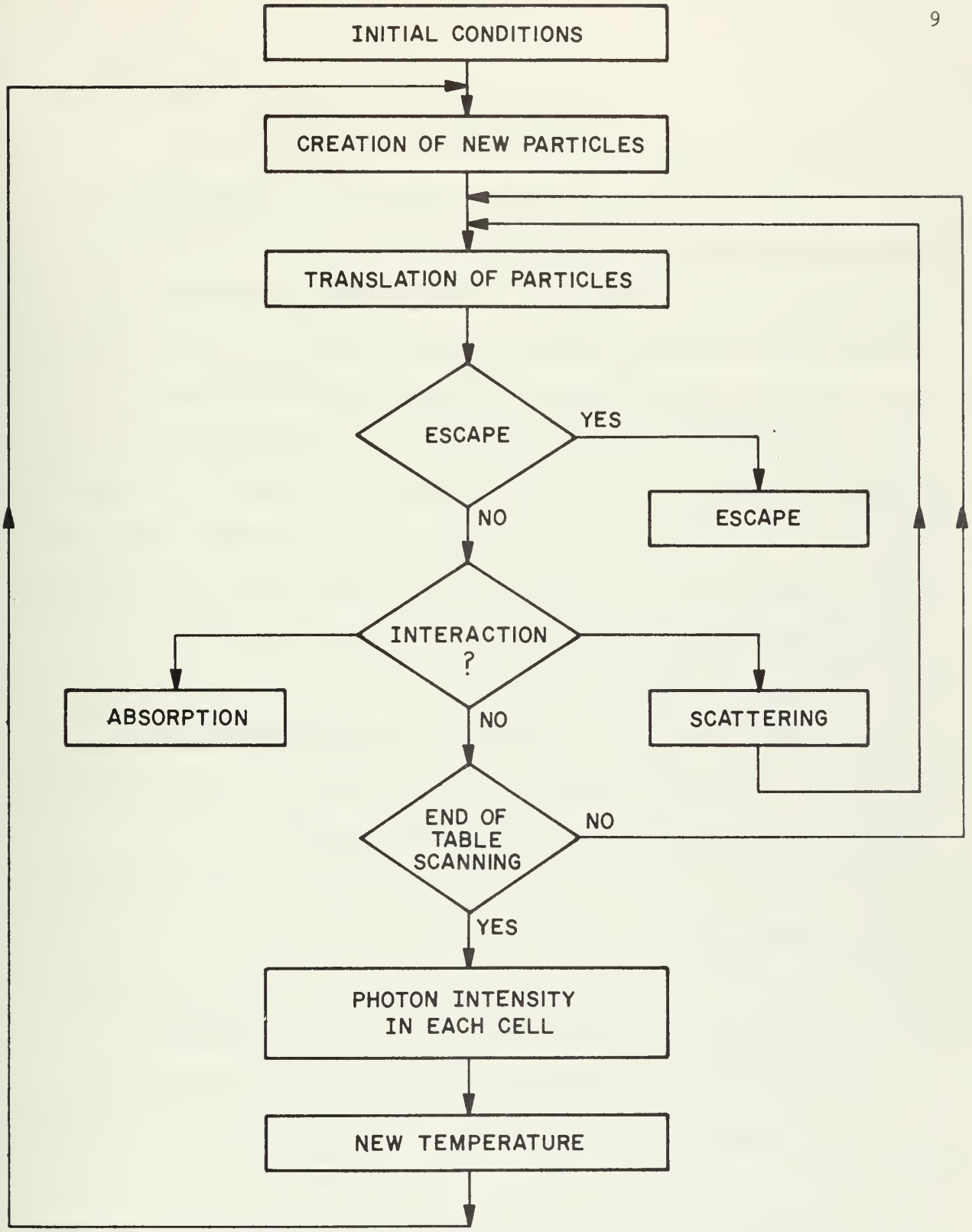


Figure 1.4 A Flow Chart of a Monte Carlo Program

2. RANDOM SAMPLING

An important part of a Monte Carlo method is the generation of good random numbers. The generation of uniformly distributed random numbers is discussed in Section 2.1. Transformation from the uniform distribution to an arbitrary distribution is discussed with some examples which are encountered in transport problems (Sections 2.2 and 2.3). In this and subsequent chapters, an emphasis is put on the application of the Monte Carlo method to the ILLIAC IV computer, and readers are assumed to have a fundamental knowledge of it (especially its architecture). (Ref. [11]). Terms such as PE (the processing element), CU (the control unit) will be used.

2.1 Linear Congruential Method

Since von Neumann's middle-square method several methods have been investigated to generate uniformly distributed random numbers. (Ref. [6]). The linear congruential method is generally accepted to be the best for digital computers for the following reasons:

(1) it is easy to calculate the subsequent random number from the updated one,

(2) only a very small space in the memory is required (unlike storing the entire or partial table of random numbers),

(3) repeated reproduction of the same sequence is possible,

(4) the period of random sequence is long (if proper coefficients are chosen).

The linear congruential method can be described by the following formula:

$$Xn+1 = (AXn + C) \bmod T \qquad (2,1)$$

where A and C are constants, Xn is the nth random number and T is also a constant. The initial value Xo is given. Xn ranges between 0 and T. $R_n = X_n/T$ ranges in (0,1). The method is called multiplicative if C = 0. Otherwise, it is called mixed congruential. T is usually chosen to be the register length of the computer (2^{48} in ILLIAC IV). A and C must be such that Xn yields good uniform random sequence. Consult reference [6] or [14] for detail, since the principle of this method is not to be presented in this paper. The method which has been adopted for the ILLIAC IV is the multiplicative congruential method (Ref. [14]) but it is also possible to use the mixed congruential method. In both methods the coefficients should be chosen to give the longest possible period of random sequence so that one sequence of random numbers can be decomposed into 64 sufficiently long subsequences. The easiest way is to divide the entire sequence into 64 subsequences and generate each subsequence in a different PE. The problem with this method is the serial correlations of random numbers across PEs or within each PE. For example, if each processor element starts with every 2^{20} random number of the original sequence then the correlations of random numbers separated by distance 2^{20} or its multiples must be closely examined, because they will be generated simultaneously in PEs and any non-negligible correlation among them could cause trouble. Nothing is known about the correlations over that long distance for the coefficients currently recommended for conventional machines. Usually serial correlations of distance up to 10 or 20 are examined. Therefore, closer attention must be paid to choosing proper coefficients.

There is a formula about the bound of serial correlation factor of any distance r (Ref. [5]):

$$\rho_r \leq \frac{A_r}{T} + \frac{1}{A_r}\left(1 - \frac{6C_r}{T} + 6\left(\frac{C_r}{T}\right)^2\right) \qquad (2,2)$$

where $X_{s+r} = (A_r X_s + C_r) \bmod T$

$A_r = A^r \bmod T$

$C_r = C \cdot \frac{A^r - 1}{A-1} \bmod T$

$$\rho_r = \lim_{n\to\infty} \frac{\sum_{k=1}^{n-r} X_k \cdot X_{k+r} - \left(\sum_{k=1}^{n} X_k\right)^2}{\sum_{k=1}^{n} X_k^2 - \left(\sum_{k=1}^{n} X_k\right)^2}$$

$A_r \simeq \sqrt{T}$ in (2,2) yields the smallest serial correlation factor ρ_r in mixed congruential methods. But it is impossible to choose A so that A_1, A_2,. . . all become the order of $\sqrt{T}$. Therefore, an actual test run in each case is necessary. Refer to a document to be published in the future for the result of the example given in Reference [14]. ($A=5^{17}$, $C=0$, $T=2^{48}$).

The method to use--multiplicative or mixed--is another problem. Statistical tests have shown that the multiplicative method is better than the mixed. (Ref. [9]). The advantage of the mixed method is that all the integers between 0 and T are covered so that sequence can be started at any number. In the multiplicative method, on the other hand, the initial random number must be chosen very carefully. The actual algorithm used for the computation on ILLIAC IV will be given in a document to be published.

2.2 Random Sampling from a Given Distribution Function

There are several methods to sample random numbers of a given distribution of uniform random numbers.

2.2.1 Rejection Method

The rejection method was proposed by von Neumann and was used in his works on neutron transport simulation. (Ref. [13]). Theoretically, this method can be applied to any type of density function. It is not efficient, however,

for the functions which have sharp maxima. In those cases many random numbers are wasted and as a consequence the number of trials increases. If a density function has a domain [a,b] and maximum value f_{max}, the rate of success is $1/(f_{max} \times (b - a))$. (Fig. 2.1). This disadvantage is disastrous if the method is applied in a straightforward way to parallel sampling, since one cannot proceed to the next step until all the PEs accept random numbers.

Suppose, for example, that there are N PEs, each of which samples a random number in parallel using the rejection method. Let p denote the probability of accepting a random number in a PE in each individual trial, i.e. $p = 1/(f_{max} \times (b - a))$. Then the probability that each PE has accepted at least one random number after r trials is

$$P(p, r, N) = \{1 - (1 - p)^r\}^N \qquad (2,3)$$

This can be shown very easily because $(1 - p)^r$ is the probability that none of the r trials are successful for individual PE.

Take Planckian case, for example. Then

$$f(x) = \frac{15}{\pi^4} \cdot \frac{x^3}{\exp(x) - 1} \quad f_{max} = .23$$

$f(10) = .8 \times 10^{-2}$ $\therefore$ $F(10) \simeq 1$ (Take b = 10 and a = 0)

then $p = 1/(23 \times 10) = .435$

with N = 64 (ILLIAC IV),

$$P(.435, r, 64) = \{1 - (.565)^r\}^{64}$$

Several curves are plotted in Figures 2.2 and 2.3. Figure 2.2 shows a rapid fall-off of the efficiency with the decrease of p.

As models become more complicated, various kinds of distribution functions will appear to determine scattering angles for high energy particles (Klein-Nishina's formula) or pair creations followed by reannihilation

etc. Each of these processes contains several branches in itself; overall efficiency is not high. Table 2.1 shows some examples taken from Reference [15]. Efficient implementation of these processes using the rejection method for the ILLIAC IV is very difficult.

2.2.2 Method of Inverse Function

If a distribution function F has an inverse function F^{-1} which is not complicated to evaluate, then a random number x can be sampled from f=F' according to the following formula:

$$R = \int_{-\infty}^{x} f(x')dx'$$

$$\text{or } x = F^{-1}(R)$$

where f is a density function, and R is a uniformly distributed random number

The advantage of this method is that all the PEs can produce random numbers simultaneously. But this advantage will be cancelled out if F^{-1} is very complicated and takes a long time to evaluate. Exponential distribution is a typical example for use of this method:

$$f(x) = \lambda\exp(-\lambda x),\ x = 1/\lambda \log (1-R)$$

where R is a sampled uniform random number .

2.2.3 Approximation of Distribution Functions by some Simpler Ones

If the inverse of distribution functions cannot be obtained explicitly, or are not simple to calculate, they are approximated by some simpler functions, or more frequently by a set of data points. The latter is essentially equivalent to the inverse linear interpolation with table lookup technique.

The method can be described as follows:

The region (0,1) on the ordinate (i.e., accumulated distribution function) is divided into N equal subintervals. Then a uniform random number R is

sampled and the interval in which it is located is checked. A random number X with required distribution f is calculated by using ordinates and abscissas of the two points which are the edges of the selected subintervals, i.e., if $F_k<R<F_{k+1}$, then $x = \frac{(F_{k+1} - R) \cdot X_k + (R - F_k) \cdot X_{k+1}}{F_{k+1} - F} =$

$$X_k(k + 1 - N\cdot R) + X_{k+1}(N\cdot R - k) \qquad (2,4)$$

The basic idea of this method is to chop the graph of a given probability distribution function f into N strips of equal area and approximate f by the average fk in each interval so that $fk \cdot (X_{k+1} - X_k)=1/N$. (Fig. 2.4). Obviously the accuracy is good near maximum point of f where the width of the strip is relatively narrow. On the other hand, it is not good in the region where f is close to zero (e.g. edge regions of Gaussian or Planckian type distributions). Figure 2.5 shows an example of Planckian distribution. (Sampling number = 10,000, N = 32).

Sometimes a density function contains a parameter μ . Consider the following example:

$$F(x,\mu) = 1/8 \{(3-\mu^2)x + (\mu^2 - 1/3)x^3\} \quad 0\leq x,\ \mu\leq 1 \qquad (2,5)$$

which is used to determine a new direction cosine x from an old one, μ, in one dimensional radiative transport problems. Interpolation in this case involves two variables, x and μ. Data is stored as a reference table; they are values $x(k,m)(0\leq k,m\leq n)$, which satisfy $F(x(k,m),\mu_m) = k/N$ $(0\leq k,m\leq 1)$. A random number x for given μ is then calculated by:

$$X(\mu) = (\mu_{m+1} - \mu)\, X(\mu=\mu_m) + (\mu-\mu_m)\, X(\mu=\mu_{m+1})$$

$$= X(k,m)(k+1-N\cdot R)(m+1-\mu\cdot N) + X(k,m+1)(k+1-N\cdot R)(\mu\cdot N-m) + X(k+1,m)$$

$$(N\cdot R-k)(m+1-\mu\cdot N) + X(k+1,m+1)(N\cdot R-k)(\mu\cdot N-m) \text{ where } \mu_m<\mu<\mu_{m+1},$$

$X(i,j) = F^{-1}(i/N,\mu_j)$ and k,m are chosen so that (R,μ) in $X-\mu$

plane is inside a trapezoid formed by four points (X(k,m),m),(X(k+1,m),m), (X(K,m+1),m+1), and (X(K+1,m+1),m+1).

More precisely, linear interpolation is first applied to the two planes $\mu = \mu_m$ and $\mu = \mu_{m+1}$ separately, then the two lines obtained as a result are used to get the ultimate interpolation line which is parallel to the x = constant plane and on which the point in question is located. Therefore, the following errors must be taken into account:

$$|E_m| \leq \max_{(x_k,\, x_{k+1})} \left|\frac{\partial^2 f}{\partial x^2}\right|_{\mu=\mu_m} \cdot (x_{k+1} - x)(x - x_k)/2,$$

$$|E_{m+1}| \leq \max_{(x_k,\, x_{k+1})} \left|\frac{\partial^2 f}{\partial x^2}\right|_{\mu=\mu_{m+1}} \cdot (x_{k+1} - x)(x - x_k)/2,$$

$$|E_m| = \max(|E_m|, |E_{m+1}|),$$

$$|E_x| = \max_{(\mu_m,\, \mu_{m+1})} \left|\frac{\partial^2 f}{\partial \mu^2}\right|_{x=x} \cdot (\mu_{m+1} - \mu)(\mu - \mu_m)/2,$$

$$E_{tot} \leq E_m + E_x \leq (h^2 M + k^2 M')/8 \qquad (2,7)$$

where

$$h = (x_{k+1} - x_k)$$

$$k = \mu_{m+1} - \mu_m$$

$$M = \max \left|\frac{\partial^2 f}{\partial x^2}\right| \quad (X_k \leq X \leq X_{k+1},\ \mu_m \leq \mu \leq \mu_{m+1})$$

$$M' = \max \left|\frac{\partial^2 f}{\partial \mu^2}\right| \quad (X_k \leq X \leq X_{k+1},\ \mu_m \leq \mu \leq \mu_{m+1})$$

Example: N = 32

$$F = \{(3 - \mu^2) \cdot x + (\mu^2 - 1/3)x^3\} \cdot 3/8$$

$$M = 3/2 \quad (0 \leq x, \mu \leq 1)$$

$$N = \sqrt{3}/6 \quad (0 \leq x, \mu \leq 1)$$

$$\therefore E_{tot} \leq 1/8 \cdot 1/1024 \cdot (3/2 + \sqrt{3}/6)$$

$$\simeq 2.2 \cdot 10^{-5}$$

This method is good when memory size is not valuable compared with the number of arithmetic operations or when $F(x, \mu)$ is very complicated to evaluate.

2.2.4 Root-finding Method

Sometimes formulae for functional iteration (such as Newton's formula) are useful to generate random numbers.

Suppose the probability density function is $p(x)$ and its accumulated distribution function is $F(x)$. If a random number R is sampled out of a uniform population in [0,1], then $F(x) - R = 0$ when solved for x, yields a random sample from $p(x)$. Several iterations of Newton's method are enough to get a satisfactory result if $F(x)$ is not too ill-behaved.

It is very important in parallel computation that all PEs get accurate values after a certain number of iterations, regardless of the distribution of Rs. (Note that all PEs have different Rs). In the ILLIAC IV, any logical decision which involves the CU (control unit) takes a long time and is desirable to avoid. The following illustrates this. To determine a scattering angle in Thomson scattering, $p(x) = 3/4\ (1 + x^2)$ $(0 \leq x \leq 1)$. Hence $F(x) = 3/4\ (x + \frac{x^3}{3})$ and Newton's formula to calculate a next approximation is:

$$X_{n+1} = X_n - \frac{F(Xn)}{f(Xn)} = 4/3 \cdot \frac{R + \frac{Xn^3}{2}}{1 + Xn^2} \qquad (2,8)$$

$X_0 = .7$ turned out to be the best as the initial value; three iterations give satisfactory results ($<10^{-8}$), whatever R may be.

In general, this method is not very efficient if evaluation of the right hand side of (2,8) requires much computation; i.e., some transcendental functions such as exponential or sine are involved. The third method and

the fourth method of this section are complimentary to each other in a sense. Note that both the root-finding and the inverse function method share an advantage over the interpolation method. They require less memory space.

2.3 Some Remarks on Random Sampling

2.3.1 Forcing (sampling exactly n samples out of N)

Forcing is one of the techniques to reduce the variance of sampling. Suppose 10% of samples out of a population which contains N elements are selected. The simplest way is to generate random numbers R and compare them with 0.1. If $R \leq 0.1$ at the Kth trial then select the Kth element. Thus at the end of N trials, 0.1N samples on the average will be obtained. But sometimes exactly 0.1N samples might be needed.

The alternative method is to keep changing the criterion at each step so that exactly 0.1N samples can be obtained at the end. This is shown in Reference [6], for example. The probability of selecting the K - 1st element is $\frac{0.1N - n}{N-k}$ if n samples have already been sampled in the K steps.

This method is applicable to the ILLIAC IV in the usual way within PEs. It is difficult, however, to apply it to select exactly n PEs out of 64, although this kind of problem may be occasionally encountered. The reason is that the process is inherently sequential in the sense that the probability to pick up the t + 1st element is dependent on the number of elements selected so far. Therefore, an efficient parallel selection is difficult.

In the case in which the sampling criterion n is large it is possible to increase the efficiency by dividing 64 PEs into several sub-groups of the same number of PEs such as 4 groups of 16 PEs, and each group

selects n/4 PEs randomly within itself. But the sampling is not free from biasing and it might cause trouble in some cases.

2.3.2 Shuffling (or random permutation of data)

Shuffling is also a task which is not easily adopted to the ILLIAC IV. Data can be shuffled in both ways--within PEs or across PEs. The former case causes no trouble and the algorithm to be stated below works in parallel, but one will encounter the same difficulty as with forcing in the latter case. There are two problems involved in the random permutation: (1) how to generate addresses to send data to and (2) how to send data actually. The first problem is equivalent to generating 64 random integers ($0 \sim 63$) each of which is different from all others. If each PE generated a random integer independently of others, the probability of success is $64!/64^{64} \sim e^{-64}$. The algorithm suggested by Knuth (Ref. [6]) is applicable, but not efficient. It states:

Suppose the data are A(1) through A(N). In the first step of the problem given above, A(i) is in the i - 1th PE and exchange of data is performed by means of the routing registers. Now, there is a register R which contains an integer. Initial value of R is N. Then a random integer S is generated which ranges from 1 to the content of R. Interchange A(S) with A(R). Decrease the content of R by 1. Generate S, interchange, . . . and so forth. This process is continued until R becomes 1.

Obviously, the method is sequential and 64 pairs of routing (forward and backward) are necessary.

The second step of the given problem (namely, sending data to a specified address) is a special case of the table lookup method, in which the one-to-one correspondence between the sources and destinations is guaranteed. The only conceivable method is to use the routing register as

"conveyer" to put all the data on and to drop the appropriate data to their destinations after each routing of distance 1. This takes 64 routings in the worst case.

Table 2.1

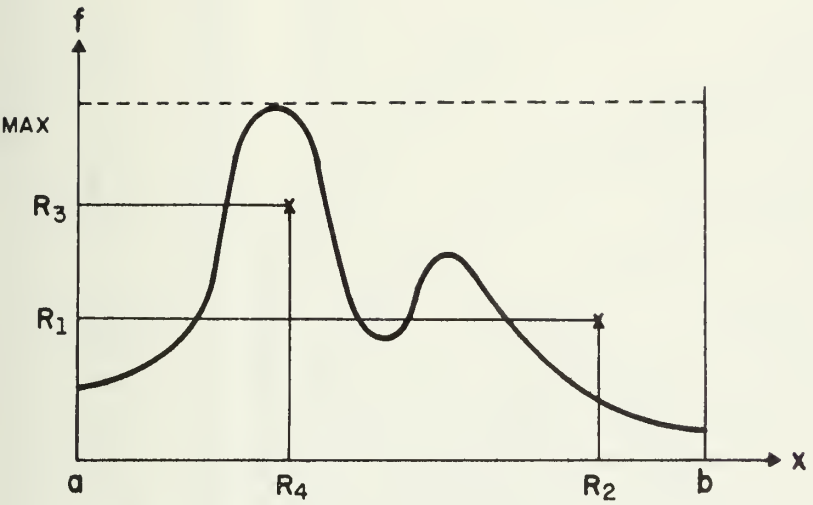
Type of Processes	$n^{(1)}$	$p = n^{-1}$ (2)	$b^{(3)}$
exponential distribution	4.3		3
$\sin\phi$ and $\cos\phi$ (ϕ: uniform)		$\frac{\pi}{4} = .785$	1
direction cosine (Covoyou)	5.42		2
direction cosine (von Neumann)	5.15		2
Compton scattering		.34~.65	3
Compton scattering (improved)		.75~.93	7
pair creation	not shown	not shown	4

Note:

(1) n : the number of generated random numbers on the average

(2) p : efficiency (reciprocal of n)

(3) b : the number of branches in a program which describes each process. The more the number of branches is, the less the PE efficiency is.



$a \leq R_1, R_3 \leq f_{MAX}$
$a \leq R_2, R_4 \leq b$
(UNIFORMLY DISTRIBUTED)
(R_1, R_2) REJECTED
(R_3, R_4) ACCEPTED

Figure 2.1 The Rejection Method

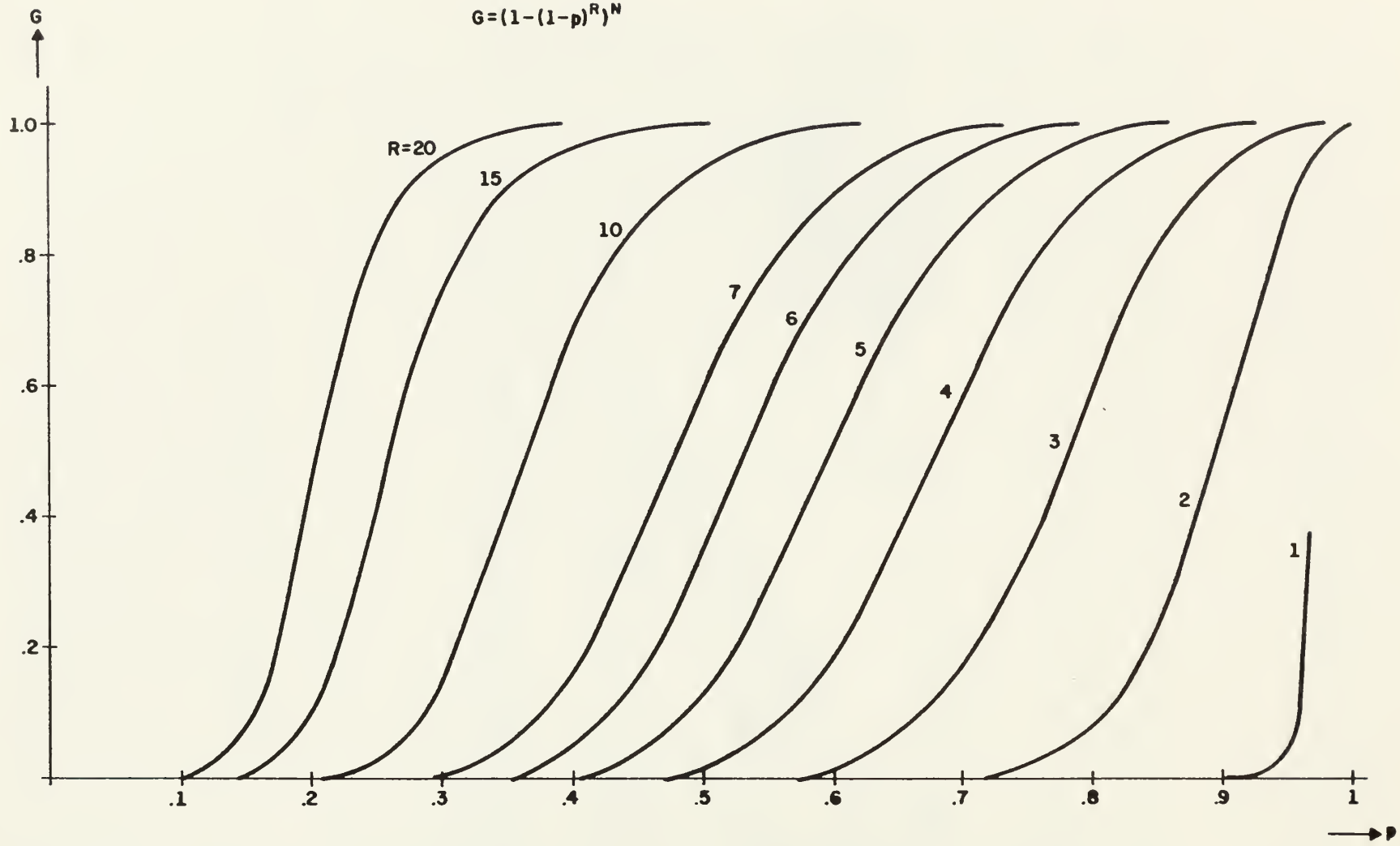


Figure 2.2 Distribution Function of the Probability of Getting All N Successful Results After R Trials (R: parameter, N=64)

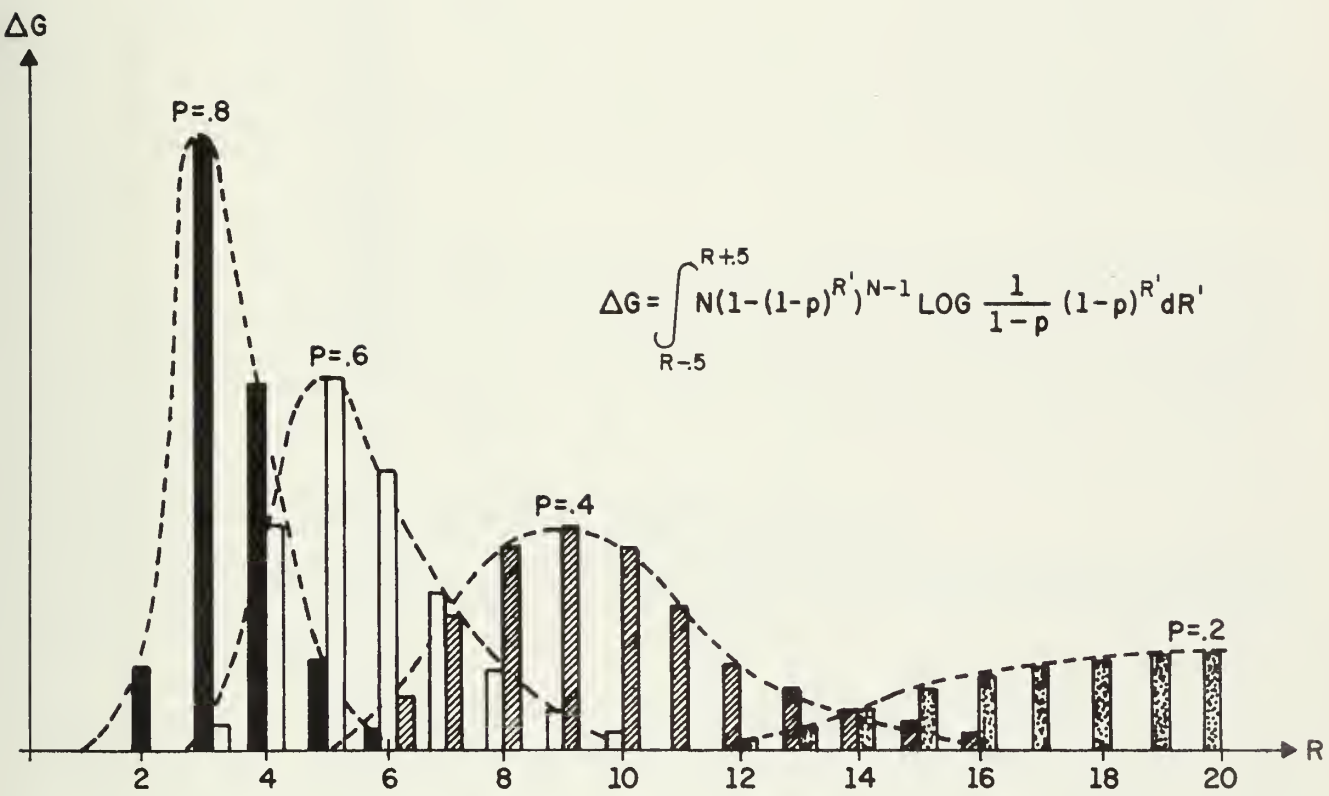


Figure 2.3 The Probability Density Function of the Number of Trials to Get All N Successful Results (p: parameter, N=64)

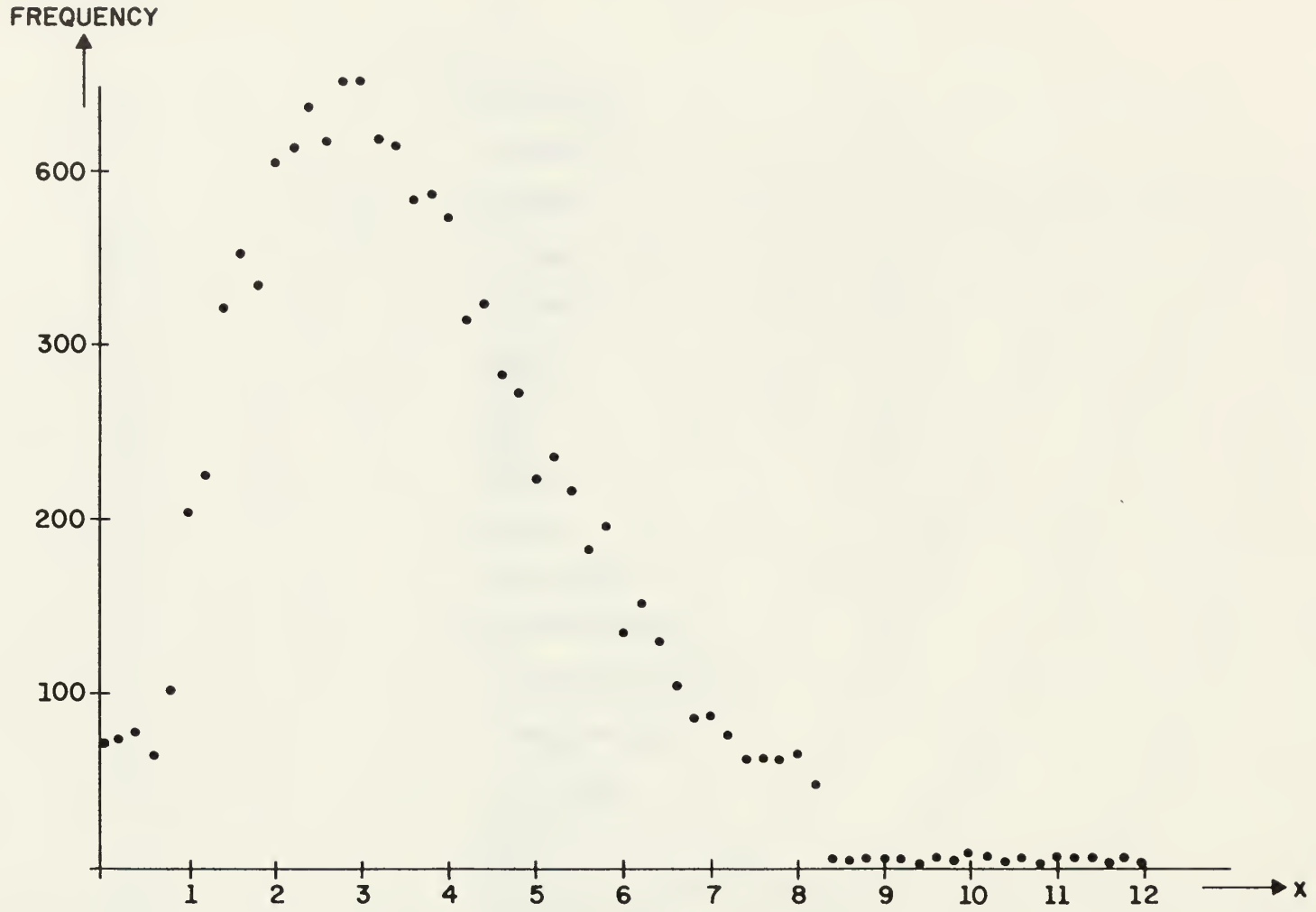


Figure 2.4 An Example of the Method in 2.2.3 Planckian Distribution

$$P(x) = \frac{15}{\pi^4} \int_0^x \frac{x^3}{e^x - 1}\, dx \quad N = 10{,}000$$

3. FUNDAMENTAL STRUCTURE OF THE PROGRAM

Both this chapter and the next chapter describe a parallel version of the Monte Carlo calculation of radiation transport. This chapter is concerned with the fundamental structure and the problem of data storage. In Section 3.1, several basic schemes are discussed and compared. The "Particle Migration Scheme" has been adopted for the problem in Chapter 1.

The actual data structure of the "Particle Migration Scheme" is described in detail in the next section. Several techniques such as conversion of real type data into integers and chain structure are also discussed.

3.1 Basic Program Structures and Their Comparison

3.1.1 Geometrical Structure of the Medium and Hardware Structures of the Computer

It is assumed in the subsequent discussions that the medium is a two dimensional rectangle. Some examples are shown in Figure 3.1. The simplest case is when the density of the medium ρ is a constant or slow-varying function of coordinates so that the table for ρ can be contained within a PE. Each PE can then cover the whole domain of the medium. In such a case it does not matter which PE processes which particle; it remains in the same PE until destroyed. The entire program becomes very simple and the efficiency depends chiefly on the number of logical branches.

If the table for ρ is too large to store in one PE (or too uneconomical), it must be distributed among PEs. In this case, the correspondence of the hardware structure (PEs and their memory) to the geometrical structure of the medium (ρ table) becomes more explicit. Let the coordinate

system attached to the medium be called the "reference coordinates". As an example, if the X coordinate of the reference corresponds to the location of PE memory and the Y coordinate to PE number then both structures match each other. (Straight storage case) As will be discussed later there are some other ways of storing ρ's, in which the correspondence becomes more obscured. As for the particles, there are two different basic points of view which consequently lead to two different program schemes.

The first scheme is to fix the reference coordinates to PEs so that particles are transferred to the neighboring PEs as they move. In other words, each PE is assigned within a certain region of the entire medium. This view is similar to the "Eulerian View" in hydrodynamics and will be called a "Particle Migration Scheme" in the following sections. The second scheme, on the other hand, is that each PE contains a certain number of particles which will be handled in it until they are destroyed. Therefore, the new data on the medium are fetched to a PE in which a particle is stored that requires them. In other words, if one's eyes are fixed on a PE (or the hardware structure) a particle stays in it and the reference coordinates move in to the particle. One confusing thing about this notion is that the motion of the reference coordinates varies with particles. This is to be called the "Coordinate Migration Scheme". The Lagrangean formulation in hydrodynamics is somewhat similar to this scheme. Comparisons of the two schemes will be made in the next sections.

3.1.2 Particle Migration Scheme

As has been discussed briefly in the beginning of this chapter, particles actually migrate to the neighboring PE as they cross cell boundaries. This is most efficiently performed by using ROUTE instruction. The merit of this scheme is that the destinations of routing are limited to

very few PEs. This makes a program work very efficiently. (Some statistical results for a simple test program will be discussed in Chapter 5). There are several ways of assigning PEs to the parts of the medium. The simplest one is shown in Figure 3.3-a. In this example, each PE is assigned with a long strip of the domain, i.e., the Y direction corresponds to processor number, whereas the X direction corresponds to the location in the memory of each PE. If the geometrical structure of the material is cylindrical and the source is located at its center, this configuration can be adopted. The advantage is that the routing distances are +1, -1, or 0 so that the program works very efficiently. In the example of Figure 3.3-a, the displacement of particles in the X direction does not involve routing. There is a disadvantage in the straight storage if many particles escape outward from the edge of the medium. Then only PE63 and PE0 will have to take care of those which escape in positive and negative y directions respectively after turning off all other PEs. This disparity on the boundary can be solved by skewing the medium as shown in Figure 3.3-b. It is obvious that the boundary is almost equally distributed but there are now 5 different routing distances possible. There is also a problem that the initial allocation of data on density becomes more complicated. (Further comparison of both straight and skewed storage will be discussed in Chapter 5).

If there is no cylindrical symmetry, the best way to distribute the boundary equally is to form 8 x 8 squares (Checkerboard storage). (Fig. 3.3-c,d). In these cases, there are 9 different routing distances each of which corresponds to a neighbor of a cell or a cell itself. If each particle carries a lot of information with it which has to be routed to its destination, the increase in routing deteriorates. The boundary problem

arises in this scheme too. In Figure 3.3-d the boundary is more equally distributed than in Figure 3.3-c.

The conclusion is that the way PEs should be assigned cells is strongly dependent on what type of problem is to be solved and cannot be determined a priori.

Reducing the variables characterizing each particle is as important as reducing the number of different route distances. This will be discussed and illustrated in detail in Section 3.2.

3.1.3 Coordinate Migration Method

In contrast to the particle migration method, particles stay in their PEs all through their lives. This method involves a table lookup problem. As time goes on, particles diffuse over the medium. They will require the data on the density stored in various PEs. Generally, this method is not applicable in practical cases. But when the table size describing the density is small enough to store entirely in a few PEs ($2 \sim 4$), then the routing distance is limited and this method is usable. The routing procedure for the "reduced density" is simpler than the particle migration method in general, because the number of data to be routed is usually smaller. As this method was not adopted in the program shown in Chapter 4 no further detail will be discussed.

3.2 Particle Migration Scheme

3.2.1 Particle Tables

Each "particle" in the table consists of such data as cell number, (N), relative coordinates within a cell, (X,Y), direction cosine, (μ), azimuthal angle, ($\sin \phi$, $\cos \phi$), energy, (E), fraction of time interval Δt, (ξ), survival distance, (τ), weighting factor, (W), and pointers. All except the last one are physical quantities associated with a particle, which

have been explained in Chapter 1. Pointers, on the other hand, are for higher efficiency of computation. Certain techniques yield less memory requirements and faster computation.

3.2.2 Packing Data and Integer-Real Conversion

The memory size problem is very serious because each PE has only 2048 words. Therefore, those previously mentioned data must be packed as densely as possible. Fortunately, GLYPNIR[1] enables any part of a word to be loaded to a register as an integer. If 16 bits are enough for an integer, 4 such data can be packed into a full word. Real numbers, however, can be only full-sized (48 bit mantissa) or half-sized (24 bit mantissa). It is possible to store a real number in the form of an integer if it has some upper limit. Such cases are: direction cosine, azimuthal angle, ($|\mu|$, $|\sin\phi|$, $|\cos\phi| \leq 1$) and relative coordinates ($0 \leq x$, $y < 1$). To accomplish this, the following rules were used: to store a real number, multiply it by 2^P ($p = 15$, for example) and store the obtained integer into some location; to read it out and compute with it, load it to a register as an integer, shift it to the left end of the mantissa part, and change the exponent part. With the combination of these techniques all the data on a particle are packed into 4 full words or less. (Fig. 3.4).

3.2.3 Chain Structure

The efficiency of computation is greatly improved by linking particles of the same characteristics (e.g. those to be scattered) to form a chain. The pointers which have been mentioned in the beginning of this chapter are for this purpose. Figure 3.5 shows how a pointer works. If particles of particular interest are not linked, all the particles in the

---

1. GLYPNIR is a higher level language for ILLIAC IV. (Ref. [8]).

table must be scanned. (Fig. 3.5-a).

On the other hand a pointer-linked table can be scanned efficiently. Suppose the table size is M. ℓ_i is the length of the chain in the ith PE and σM is the expectation value of ℓ_i (i = 0, 1, . . .63). Then the number of steps to scan the table is $\angle = \max_{0 \le i \le 63} \ell_i$

The distinction between the linked and the unlinked is more obvious if the entire table (M) consists of several, say N, subsets and any particle belongs to one of them and only one of them. The number of steps is N x M to scan the unlinked table, (because the table must be scanned N times), while the number of steps to scan the linked table is:

$$\angle = \sum_{I=1}^{N} \max_{0 \le i \le 63} (\ell_i (I)) \le N \times M$$

See appendix 1 for mathematical derivation and an example. An illustrated example is also given in Figure 3.6.

Pointers can be either included in particle tables or separated from them. The first method is good if there are only a few pointers. (Such is the case of the program listed in this paper). But the second method is better if there are many ($\sim$8) pointers associated with such particles; they need not be routed when a particle crosses cell boundary.

Suppose that every processor contains 100 particles each of which consists of 4 words. The pointer must take values 0, 4, 8 . . .396 (<512). Therefore, 8 pointers can be packed into a full word (64 bits). (Note that 8 bits are enough for each of them).

3.2.4 Material Table

The material table stores data such as energy (temperature), density, specific heat of 64 x 32 cells, besides tables of absorption and scattering coefficients. Each cell is assigned with two full words and data are packed

in the same way as before, some as integers, some as real numbers.

If energy of newly created particles at each time step is to be sampled from a Planckian population, its interpolation table must be stored too.

3.2.5 I/O

Because of the limitation of memory size, only a portion of large particle tables can be stored in the PE memories. Therefore, they must be stored in disc memory after being processed sufficiently and a new particle table must be brought into the PE memories. It is not efficient, however, to stop processing while I/O is being done. A better way is to divide each memory into three large blocks titled, NEW, OLD, PROCESS, which corresponds to the area into which a new particle table is being read, the area from which a particle table is stored into disc and the area of currently processed particle tables, respectively. NEW becomes PROCESS, PROCESS becomes OLD, OLD becomes NEW in the next phase and so on. (Fig. 3.7).

3.2.6 Total Capacity of Particles

The capacity of particles in the simple case when the medium consists of 64 x 32 cells can be calculated by taking all the above things into consideration:

$$\overbrace{3 \times 4 \times N}^{\text{Particles}} + \overbrace{32 \times 2 + E}^{\text{Medium}} + F \leq 2048$$

where

N : the number of particle table entries

E : table for random sampling (Planckian) + table for absorption and scattering coefficients ≃ 50 (non-monochromatic)
0 (monochromatic)

F : instructions and GLYPNIR system ≃ 200

$\therefore$ N = 1734/12 ≒ 144

64N ≒ 9200

Therefore, about 9200 entries are available for particles.

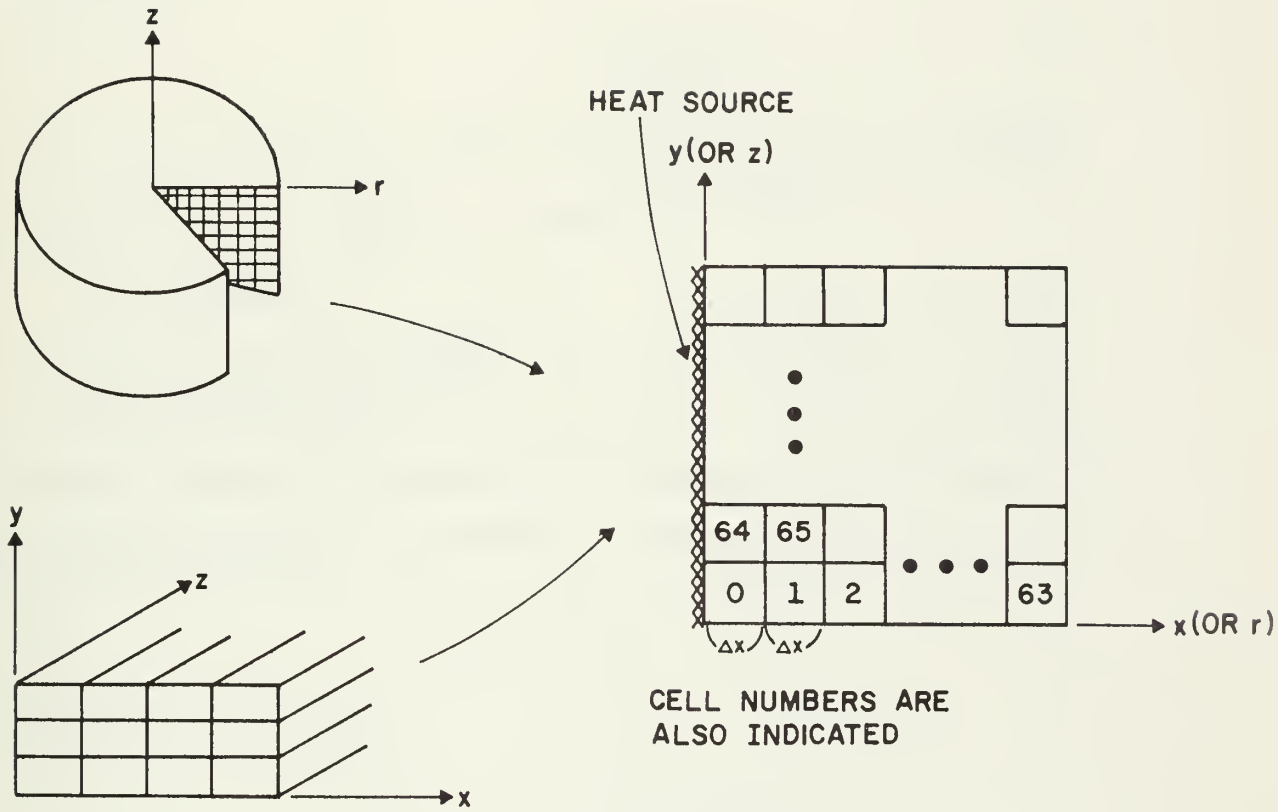


Figure 3.1 Some Examples of Geometrical Structure of the Medium and the Corresponding Cell Number Assignment

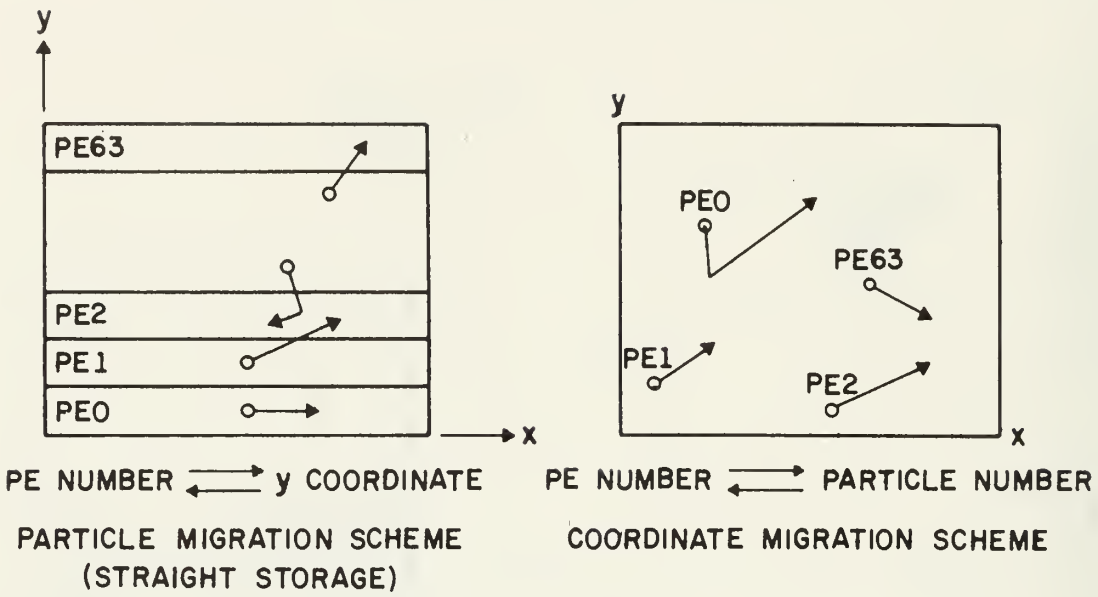
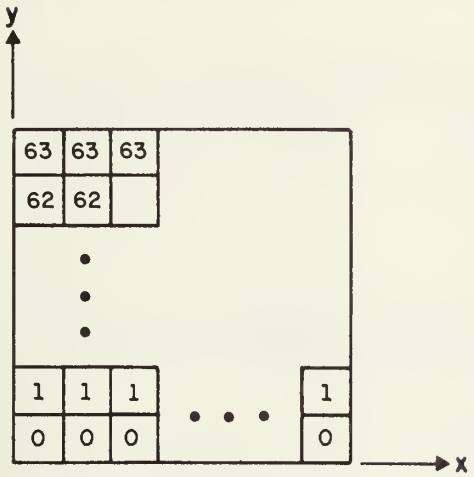
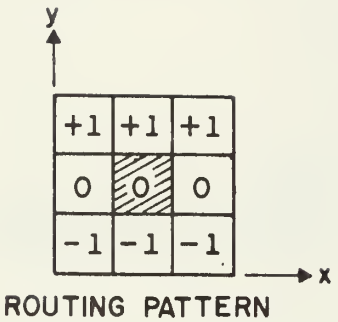


Figure 3.2 Comparison of Two Different Schemes

(a) STRAIGHT STORAGE



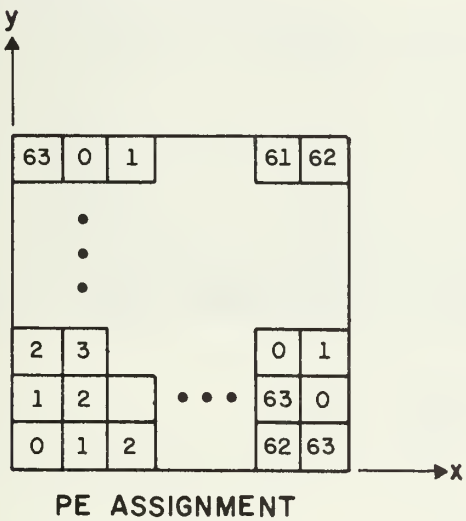
PE ASSIGNMENT



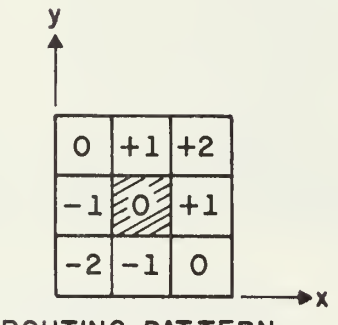
ROUTING PATTERN

(ALL NUMBERS ARE ASSIGNED
ACCORDING TO FIG. 3.1)
PE NO. = CELL NO. DIV 64

(b) SKEWED STORAGE



PE ASSIGNMENT



ROUTING PATTERN

PE NO. = (CELL NO. DIV 64 + CELL NO. MOD 64) MOD 64

Figure 3.3 Some Examples of PE Assignment
in the Particle Migration Scheme

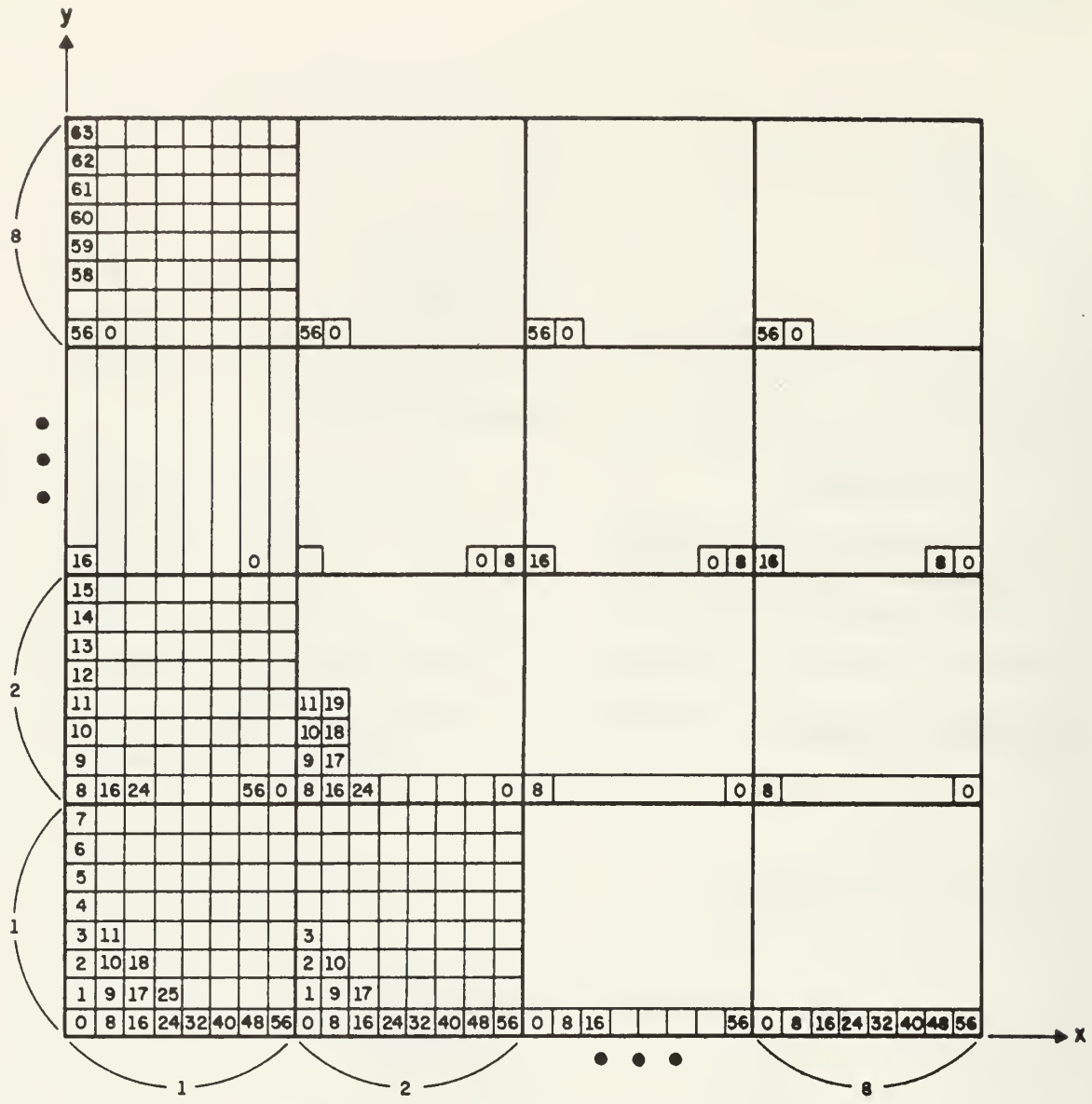


Figure 3.3-c Some Examples of PE Assignment in the Particle Migration Scheme (continued) - Straight Checkerboard Storage

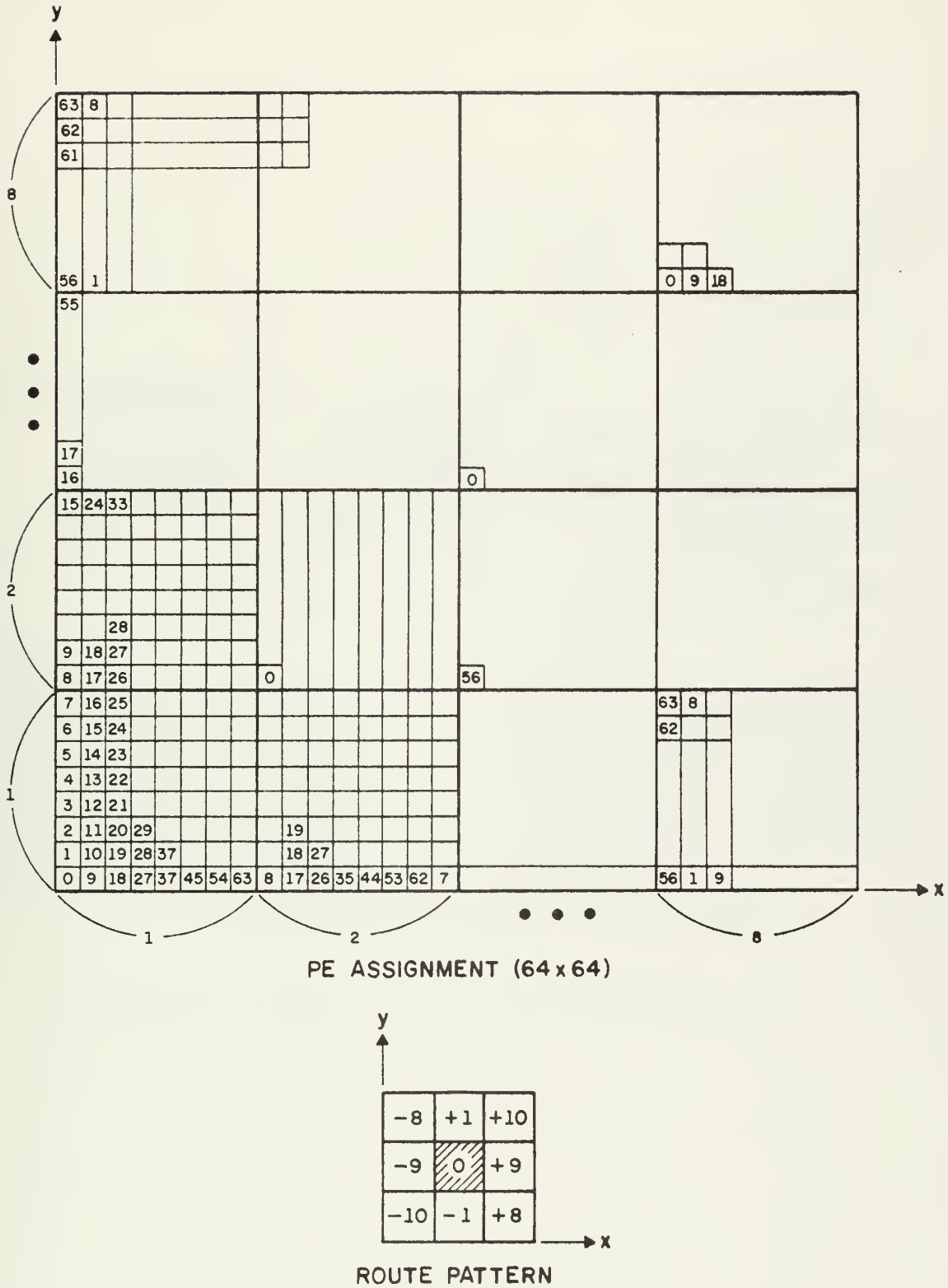
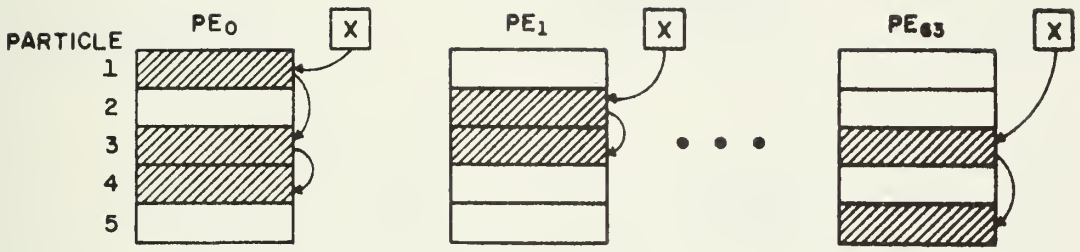


Figure 3.3-d Some Examples of PE Assignment in the Particle Migration Scheme (continued) - Skewed Checkerboard Storage

TNB+0	CELL NO.	POINTER 1	x	y
TNB+1	SIN ϕ	POINTER 2	μ	COS ϕ
TNB+2	$\xi\Delta t$		ENERGY	
TNB+3	WEIGHT		τ	

Figure 3.4 An Example of Particle Table
(used in the test program in Appendix 3)



(a) WITHOUT CHAIN

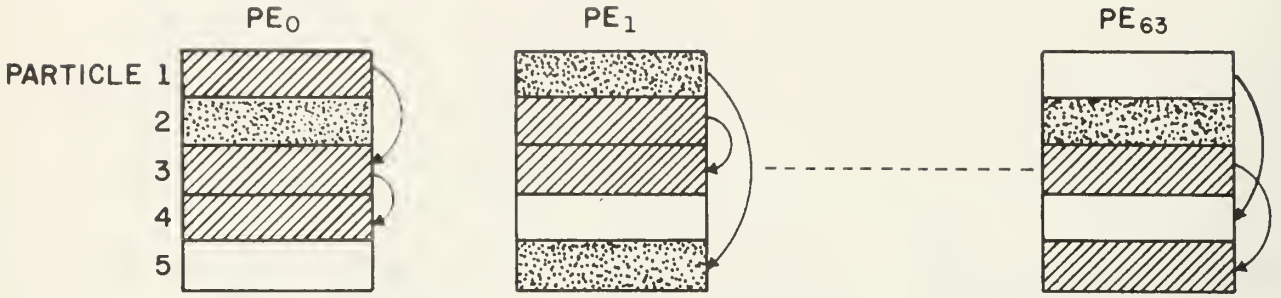
STEP	PE_0	PE_1	• • •	PE_{63}
1	ON	OFF		OFF
2	OFF	ON		OFF
3	ON	ON		ON
4	ON	OFF		OFF
5	OFF	OFF		ON

(b) WITH CHAIN

STEP	PE_0	PE_1	• • •	PE_{63}
1	ON	ON		ON
2	ON	ON		ON
3	ON	OFF		OFF

MAX. LENGTH = 3

Figure 3.5 Comparison of a Chain-linked Table and a Usual Table



TYPE	NO. OF STEPS	PE			
	1	#1	#2	----------	#3
	2	#3	#3	----------	#5
	3	#4	OFF	----------	OFF
	4	#2	#1	----------	#2
	5	OFF	#5	----------	OFF
	6	#5	#4	----------	#1
	7	OFF	OFF	----------	#3

Figure 3.6 Another Example of Chain-linked Table

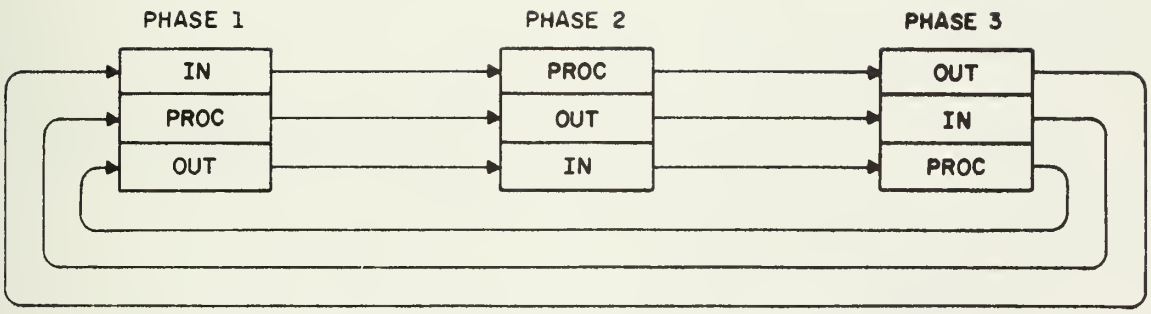


Figure 3.7 Three Phases in I/O

4. A TEST PROGRAM FOR ILLIAC IV

This chapter is concerned with the description of the program which was written in GLYPNIR (Ref. [8]) and was run on simulator SSK. (Ref. [17]). It is given in Appendix 3. This program was written to test the feasibility of parallel computation for this problem and also to serve as an example for more detailed programs.

The simplifications which were made for this test program were discussed in Section 1.3. A flow chart is in Figure 4.1.

The whole program consists of about 20 small subroutines, names of which are given in Table 4.1.

4.1 Description of Major Subroutines

DATALOAD--All the numerical quantities which are necessary to initialize the programs are read in. These are:

(1) initial random numbers

(2) the length of one time step (Δt)

(3) mass absorption coefficient (σabs)

(4) mass scattering coefficient ($\sigma scatt$)

(5) density distribution

(6) specific heat distribution

(7) initial temperature distribution

CHAIN--This subroutine is to link the table entries of some common characteristics (such as scattering, routing, etc.) to form a chain. It was briefly mentioned in Chapter 3. It functions as follows: Suppose the Nth table entry is to be linked to the chain for scattering. First, the pointer part of the Nth entry is replaced by the contents of a register which always keeps the location of the edge of the

chain for scattering. Then N (which is the most up-to-date location of the edge) is stored into the register.

Due to the necessity for handling two different pointers by one subroutine, indexing facility is also included. Namely, either of two pointer parts of a particle table entry can be specified by changing an input parameter Q. Q = 0 for scattering and vacancy, whereas Q = 1 for routing.

CVTRL--This subroutine divides a 15 bit integer by 2^{15} without using division. It was also discussed in Chapter 3.

RNDX--This is a random number generator which generates uniformly distributed random numbers by the linear congruential method. See Chapter 2 for detail. Initial random numbers are read into PEs at the beginning of the program.

ENERGYBALANCE--This subroutine calculates the energy to be emitted by radiation in each cell. The total radiative energy is summed up to assign the number of particles in the next subroutine.

PARTICLEASSIGN--The number of new particles in each cell of the medium is determined. It is proportional to the radiative energy in the cell. To avoid the overflow problem of particle table, only the two-thirds of the available vacancy in PE is used to create new particles. The energy per "particle" is merely (the radiative energy per cell)/(the number of particles in the cell). That is, the energy is monochromatic.

The whole program has been made so that the radiation process can be extended in the future to non-monochromatic case. The weighting factor in the particle table, for example, is meant for this purpose. If the radiation is not monochromatic, tables for absorption and scattering coefficients as well as data for Planckian distribution must be

stored in PE memory.

BIRTH--New particles are created in this subroutine according to Section 1.3.

TRANSLATE--This subroutine has two functions. One is to translate particles to their new locations. The other is to detect those particles which cross the cell boundaries and those which escape from the medium.

ABSORB--This subroutine is called when a particle is absorbed by the medium. The particle is destroyed and its table entry is linked to the VACANCY chain.

ESCAPE--This subroutine is called when a particle escapes from the medium. The particle is also destroyed, its table entry is linked to the VACANCY chain, and ESCAPE counter is incremented by its weight.

BRANCH--This subroutine calls TRANSLATE, decides interaction types of particles, then calls ABSORB, ESCAPE to process particles. Those which are to be scattered are not processed here but linked to form a chain. This is because the scattering process is very long and it is desirable to process them in parallel at a later time. Those particles which cross the cell boundaries toward the y-direction are also linked for routing. This subroutine is called both in the main program and SCATTER.

ANGLE--New sets of angles after Thomson scattering are calculated. The algorithm to sample the scattering angles was described in Section 2.2. New angles are then obtained by using spherical trigonometric formulae in Section 1.1.

SCATTER--Particles in the chain for scattering are processed after the table scanning is finished in the main program. This subroutine

calls ANGLE, BRANCH, STORE, and functions almost in the same way as the main program does to scan the table.

ROUTE--Particles in the chains for routing are actually transferred to their destinations in this subroutine. Index K = 1 specifies the routing to the right and K = -1 specifies the routing to the left. The vacancy which has just been made is linked to the VACANCY chain.

TALLY--This subroutine is used to sum up the intensity of radiation in each cell. The summed intensity is necessary in ENERGYBALANCE to calculate the new temperature.

4.2 Main Program

The main program contains all the global variable and subroutines and calls the subroutines in the sequence given in Figure 4.1.

The main program starts with the reservation of blocks for PCPOINTERS. The constants and counters are initialized. All the particle table entries are linked as vacancy. A large loop for iterations is entered, one step of which corresponds to the time interval Δt. The radiative energy computed in ENERGYBALANCE decides the particle distribution in PARTICLEASSIGN. BIRTH follows it to create new particles in the vacancy. Then the table scanning starts, (this is a sequential scanning from the top to the bottom), during which the coordinates of particles are updated, the types of interactions are determined, and so on. After all the entries of the particle table are scanned, the main program proceeds to the scattering process SCATTER, which is followed by ROUTE and TALLY. This completes one time cycle and the intermediate results may be printed out by TABLEPRNT. The entire program consists of more than 800 cards in source code (including comment cards) and is compiled into more than 5,500 instruction syllables

of assembly codes. About 220 words of PE memory in every PE are occupied by these instructions.

Table 4.1

Name	Function	Type	Called By
ABSORB	Absorption		BRANCH
ANGLE	New Scattering Angles		SCATTER
BIRTH	Creat New Particles		MAIN
BRANCH	Interaction Type		MAIN
CHAIN	Table Link		PARTICLEASSIGN, BIRTH, BRANCH, SCATTER, TRANSLATE, MAIN
COS	Cosine	PREAL	BRANCH
CVTRL	Integer→Real	PREAL	BRANCH
DATALOAD	Initial Data		MAIN
DSTBND	Distance to Boundary		BRANCH
ESCAPE	Count the Number of Escapes		BRANCH
ENERGYBALANCE	New Energy		MAIN
EXP	Exponential	PREAL	
LN	Natural Log	PREAL	BIRTH, SCATTER
MOD	Mod (64)	PINT	BIRTH, TRANSLATE
PARTICLEASSIGN	Number of Particles to be Created		MAIN
RNDX	Random Number	PREAL	BIRTH, ANGLE
ROUTE	Routing to Neighbors		MAIN
SCATTER	Scattering		MAIN
SIN	Sine	PREAL	BIRTH, ANGLE

Table 4.1 (con't)

Name	Function	Type	Called By
SQRT	Square Root	PREAL	BIRTH, ANGLE
STORE	Store to Particle Table		MAIN, SCATTER
TABLEPRNT	Print out the Contents of Tables		MAIN
TALLY	Intensity of Radiation in each Cell		MAIN

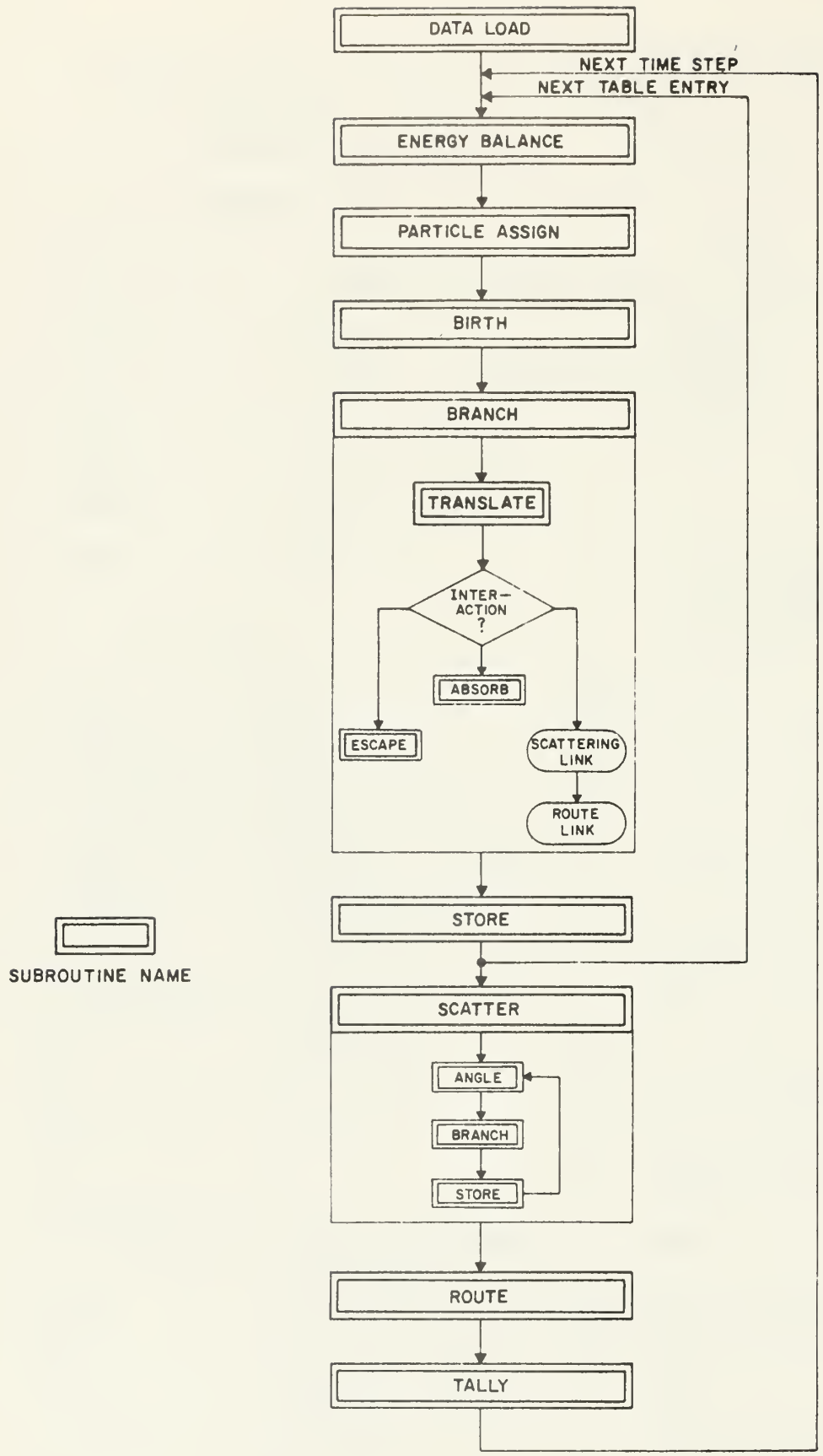


Figure 4.1 A Flow Chart of the Test Program

5. SOME COMPUTATIONAL RESULTS

Some of the computational results are discussed in this chapter. Since the simulator works very slowly, it is not practical at the present time to run the whole program for many time steps. Therefore, emphasis was put on the statistical investigation of PE efficiency aspects. Some factors investigated here are:

(1) cell size vs velocity

(2) effect of multiple scattering

(3) effect caused by using modified probability density function for the survival distance (τ)

(4) effect of skewed storage

All the test programs have been run under the following assumptions:

(1) the material is homogeneous

(2) no absorption is involved

(3) scattering cross section is a constant (σ)

(4) scattering is isotropic

(5) mesh size is 64 x 8 except one case (64 x 64)

5.1 Determination of Cell Size (ΔX) and Time Interval (Δt) and Their Relation to the Rate of Boundary-Crossing

It is important to know how many particles involve routing. This rate is predicted to be dependent on $v\Delta t/\Delta X$. Figure 5.1 shows some computational results. Sixteen particles are created in each PE at the beginning of the time interval Δt. Scattering cross section is taken as a parameter. The rate grows quite linearly with $v\Delta t/\Delta X$. This means that $v\Delta t/\Delta X$ does not affect the efficiency as far as routing is concerned; if Δt is doubled then the number of time steps is halved. The rate decreases somewhat with the

increase of σ because some of the particles which have crossed boundaries are scattered back to their original cells.

The cell size ΔX must be determined uniquely by how the density (ρ) of the medium varies with position. If ρ changes very slowly, each PE can cover a wider range (i.e., large ΔX) and vice versa.

Δt has an upper limit; no particle can travel through two or more cells in one time step, i.e., $v\Delta t \leq \Delta X$. Within this limit, Δt can be chosen arbitrarily. Small Δt will bring more accurate results on the temperature and the intensity, but the number of time steps increases. Large Δt, on the other hand, will bring more crude results in shorter time. In this case, more routing is involved in each time step. Refer to Section 5.5.4 for a related topic.

5.2 Effect of Multiple Scattering

If a particle experiences more than one scattering in Δt, the efficiency of parallelism obviously decreases. Therefore, it is important to see the relation of multiple scattering to scattering cross section σ and $\ell = V\Delta t$. The process is simply a Poisson process if σ is constant. Therefore, the probability that a particle is scattered n times while it travels the distance ℓ is: $Pn(\ell) = \frac{(\sigma\ell)^n}{n!} e^{-\sigma\ell}$. (Derivation is in Appendix 2). Figure 5.2 shows the theoretical curve (solid lines) and the result of computation. It is obvious that the number of multiple scattering increases with ℓ. At $\ell=1$ about 20% of the particles are scattered twice or more. Multiple scattering is treated just as usual scattering in the program described in Chapter 4: if a particle is re-scattered after a scattering, then it is linked again to the top of "scatter chain". Therefore, it is processed together with some others which are located in different PEs and are to be scattered for the first time.

It is not a good idea to enable PEs for only the second scattering after each scattering, then to switch to the third and so on. . . . Statistical fluctuation would work in the most unfavorable way.

5.3 An Effect Caused by Modifying a Probability Distribution Function for Survival Distances of Particles

An effort was also made to modify the probability density function for survival distance to reduce the rate of multiple scattering. One method is simply to cut off the density function below some value ℓ=a. This results in shifting all Pn(x)s ($n \geq 1$) to the right by a with the exception that Po(x)=1 ($x \geq a$). If a is chosen to be $v\Delta t$, then there will be no multiscattering at all. But this is too unnatural. The second method investigated gives a linear growth up to x=a and exponential decay. (Fig. 5.3). Starting at x=0, each curve seems to be located between the one sampled from $P(\tau)=e^{-\tau}$ and the one from $P(\tau)=e^{-(\tau-a)}$, but it soon exceeds the former. There is an unusual increase due to the sharp peak in the function $P(\tau)$ at x=a. For smaller x, however, the rate of scattering is reduced. Some order of the Erlang distribution will give a smoother result.

5.4 Skewed Cell Boundaries Vs Straight Cell Boundaries

It was stated in Chapter 3 that the skewed cell boundaries smooth out the distribution of the number of particles which escape from the medium. The ideal case is when the mesh size of the medium is 64 x 64; every PE has the same number of edges. (Fig. 3.3-b). This scheme, however, increases routing. Therefore, it is necessary to check which method is better. Several tests have been made to compare these two schemes.

The mesh sizes for testing were 64 x 8 and 64 x 64; in each case the same data were interpreated in two ways; skewed case and straight case. Results are shown in Table 5.1 and Figures 5.4, 5.5. The conditions

for numerical experiments are the same as before: isotropic scattering, constant σ , and each particle travels for the time Δt. $\Delta x=1$. σ and velocity are specified in each case.

5.4.1 64 x 8 Cells

Data were taken for $\sigma=.4$, velocity=1, and $\sigma=.2$, velocity=.5. Ten particles were created in each cell at the beginning of a time step and they were all destroyed at the end of this time step. This process was repeated 5 times and the contents of the counters were accumulated. Therefore, effectively 40 particles were examined in each cell (not in each PE). Unnecessary subroutines were removed. In Figures 3.3-a and 3.3-c, there is a peak which is due to the PEs corresponding to the edges of the medium. These peaks were fairly smoothed out in the skewed cases shown in Figures 3.3-b and 3.3-d. As for routing, the distributions turned out to be Gaussian-like in all examples. Obviously, more routing is involved in the skewed storage. Also shown are the distributions of the double routing in skewed storage. That is, the number of particles which have to be routed twice to get to their destinations. (Fig. 3.3-a).

Table 5.1 is the summary of these numerical experiments. The numbers "max-min" in the first column of each experiment mean the maximum and minimum numbers of particles among PEs, respectively, which escaped from the medium or crossed the cell boundaries. "PE average %" means the average percentage of enabled PEs. This was obtained from:

$$\text{PE average \%} = \frac{\sum_{PE=0}^{63} (\text{counter (PE)})}{\text{max} \times 64} .$$

For example, (49-4, 24%) in the first row means that there were 49 particles escaped from one PE whereas there were only 4 particles from another PE. All others are between these two values. Therefore, 49 steps are taken to

process these particles in parallel, and 24% of PEs are enabled on the average.

5.4.2 64 x 64 Cells

Data were taken for σ=.4, velocity=1, and σ=.2, velocity=.5. Only the first example is shown in Figure 5.5. Only one particle was created in each cell at the beginning of a time step and they were destroyed at the end of this time step. This process was repeated 5 times so that 5 particles were examined in each cell effectively. Obviously the number of escapes is smoothed out across PEs. The lower half of Table 5.1 can be read in the same way as described in Section 5.4.1. The conclusion is that the skewing does not always bring more favorable results than the ordinary storage.

It is true that the particles that escape from the medium are very well distributed across PEs, but the increase of the number of routing is so big that it cancels out this merit. Therefore, the skewing is better than the ordinary storage only in limited cases such as when ESCAPE takes a much longer time than ROUTE.

5.5 Concluding Remarks

The program which is described in this document is a very simple one and expandable to more complicated cases. It is inevitable, however, that the efficiency goes down as the program becomes more complicated.

There are several problems which are left to be solved.

5.5.1 Problem of Table-Overflow and Normalization

As was discussed in Chapter 3, the particle migration scheme has been adopted to obtain higher efficiency in parallel computation. As a consequence, however, particles must be physically transferred to other PEs. If many particles flow into one PE, it will not be able to accept all of them. To encounter this problem, certain number of table entries must be

saved in PEs for those visitors.

Cells must be well distributed among PEs so that the outgoing particles cancel the incoming particles on the average; if many particles flowed into one PE, it would not be able to accept all of the particles. This is an extreme case, but a certain number of table entries still must be saved in PEs as buffers for those visitors. There is a technique called "Russian Roulette" in which incoming particles are rejected with probability p and accepted with the probability 1-p. In the latter case, the weighting factors are multiplied by 1/(1-p). This is one of the typical techniques to reduce variances. Special care must be taken, however, when it is used.

5.5.2 Problem of Word Size (alternative format)

When the program was originally implemented, an emphasis was put on the capacity of the particle table. Consequently, only 16 bits were assigned to those words which range in [-1,1]. But if more accuracy is needed on those quantities, three of them can be packed into one full word. In this case, each word has 21 bits, which is comparable to the fraction part of a half-word (24 bits). An example of word format is given in Figure 5.6.

5.5.3 About the Flexibility of Logical Network Which Interconnects PEs

In the problems which involve probabilistic nature, it is highly desirable that the interconnection of PEs be as flexible as possible. A good example is the table look up problem which was discussed in Chapter 2. If the interconnection logic is programmable by users it will be much easier to handle more complicated problems. This may be one way parallel computers should be organized.

5.5.4 Variable Contraction Ratio of Cell Sizes

It is possible to change the cell sizes in X direction, Y direction, or both. Take Y direction, for example. If the density gradient is not

uniform in Y direction, it is better to adjust ΔY so that it covers a wider range when the density varies slowly and a narrower range when the density varies rapidly in that direction. The reduction ratio, α_i, in each row of cells is stored as a PE variable and the displacement of particles in Y direction is modified as

$$Y \leftarrow Y + \alpha_i \, V \, \Delta t \, \sqrt{1-\mu^2} \cdot \sin \phi$$

when the particles are in the ith row of cells.

Survival distance and the displacement in X direction are not changed in this example.

Table 5.1

		σv	σ = .1, v = .5		σ = .4, v = 1	
			Max. - Min.	PE average %	Max. - Min.	PE average %
64x8	STRAIGHT	Escape	49 - 4	24%	90 - 9	23%
		R. Route	53 - 0	72%	94 - 0	73%
		L. Route	52 - 0	72%	91 - 0	78%
	SKEW	Escape	26 - 5	53%	45 - 9	46%
		R. Route	69 - 44	85%	103 - 59	80%
		R. Dbl. Route	9 - 0		22 - 6	
		L. Route	70 - 43	84%	103 - 58	82%
		L. Dbl. Route	8 - 0		21 - 7	
64x64	STRAIGHT	Escape	35 - 0	5.7%	62 - 0	6%
		R. Route	53 - 26	71%	85 - 50	75%
		L. Route	50 - 27	74%	89 - 52	74%
	SKEW	Escape	5 - 0	40%	10 - 0	44%
		R. Route	82 - 46	74%	93 - 65	85%
		R. Dbl. Route	10 - 0		23 - 4	
		L. Route	76 - 36	83%	104 - 60	79%
		L. Dbl. Route	9 - 0		22 - 6	

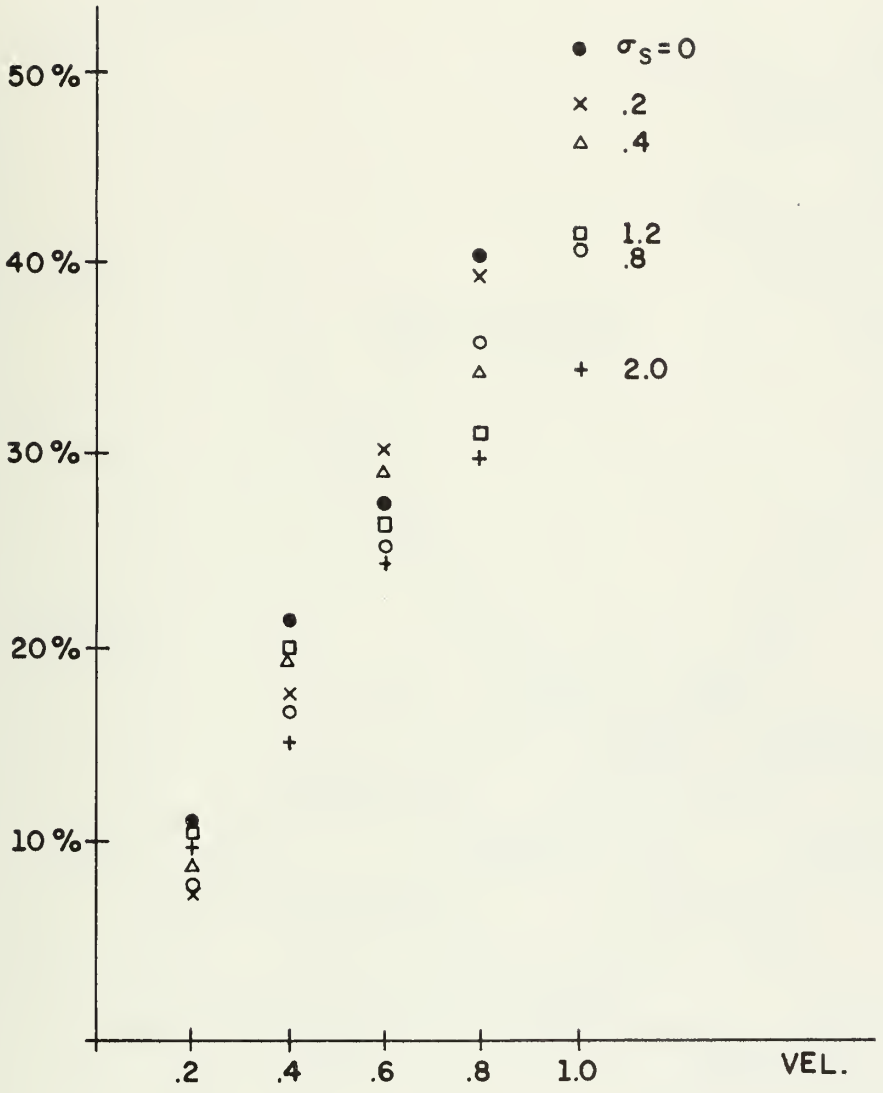


Figure 5.1 Percentage of Particles Which Have Crossed Cell Boundaries Vs Velocity
s: parameter (isotropic scattering)
Δx=1, Δt=1 1024 particles

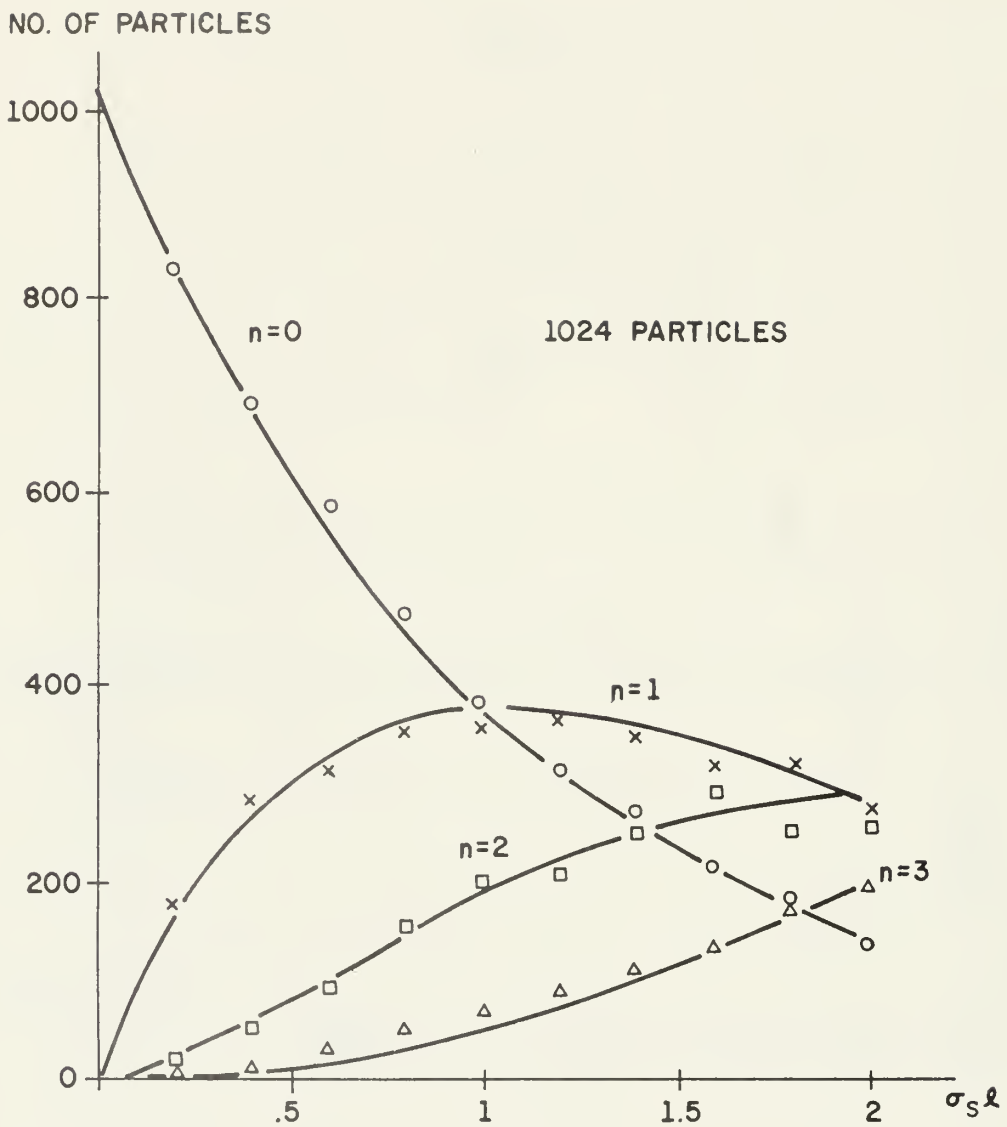


Figure 5.2 Multiple Scattering During
$\Delta t=1$ vs $\sigma_s \ell = \sigma_s v \Delta t$ (isotropic scattering)
(Solid curves are $Pn(\ell) = \frac{(\sigma_s \ell)^n}{n!} e^{-\sigma s \ell}$ n=o,1,2,3)

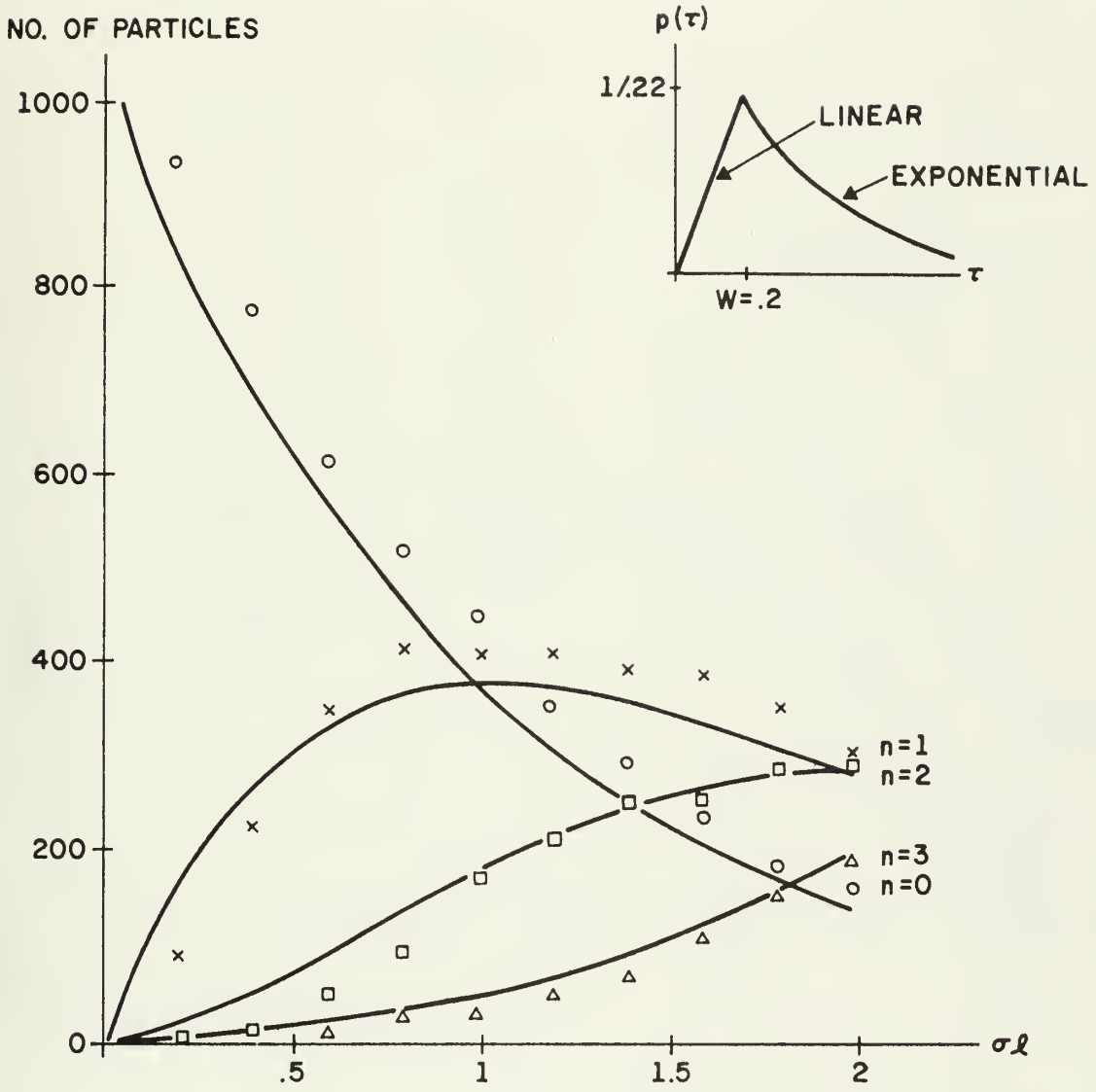


Figure 5.3 An Example of the Modified $P(\tau)$ with W=.2 (above) and the Computational Results (below) (Solid Curves are $Pn\ (\ell) = \frac{(\sigma\ell)^n}{n!} e^{-\sigma\ell}$ n=0,1,2,3)

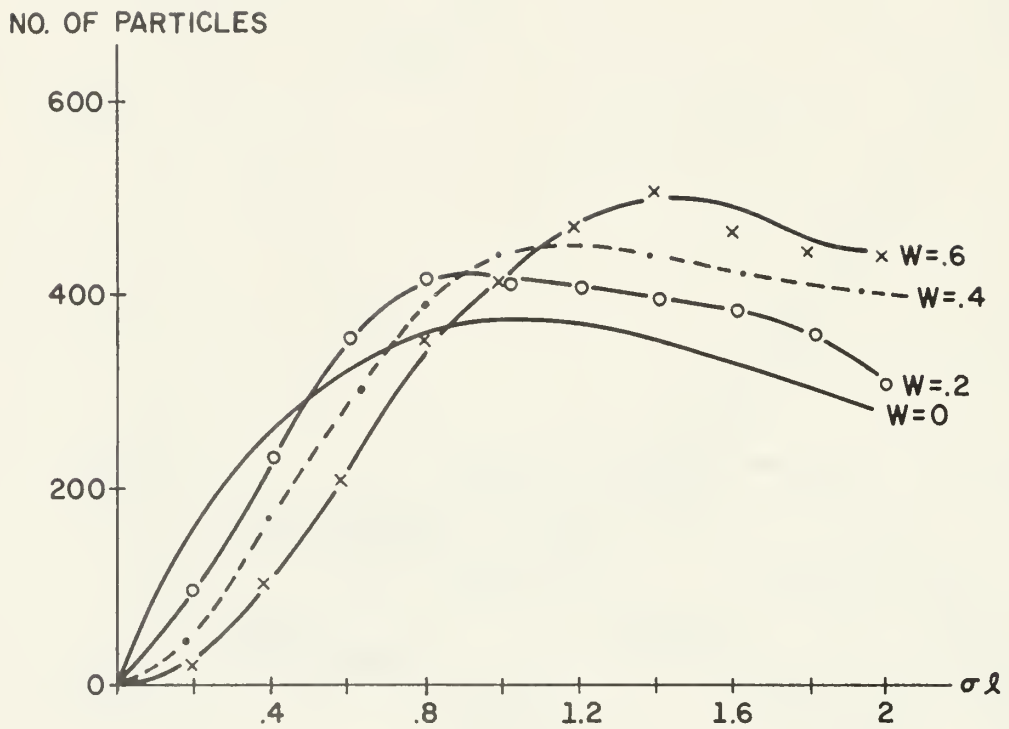


Figure 5.3 (continued) Comparison of Several P(τ)s
with Different Ws (W=0,.2,.4,.6)
(Only the Results for N=1 are Displayed)

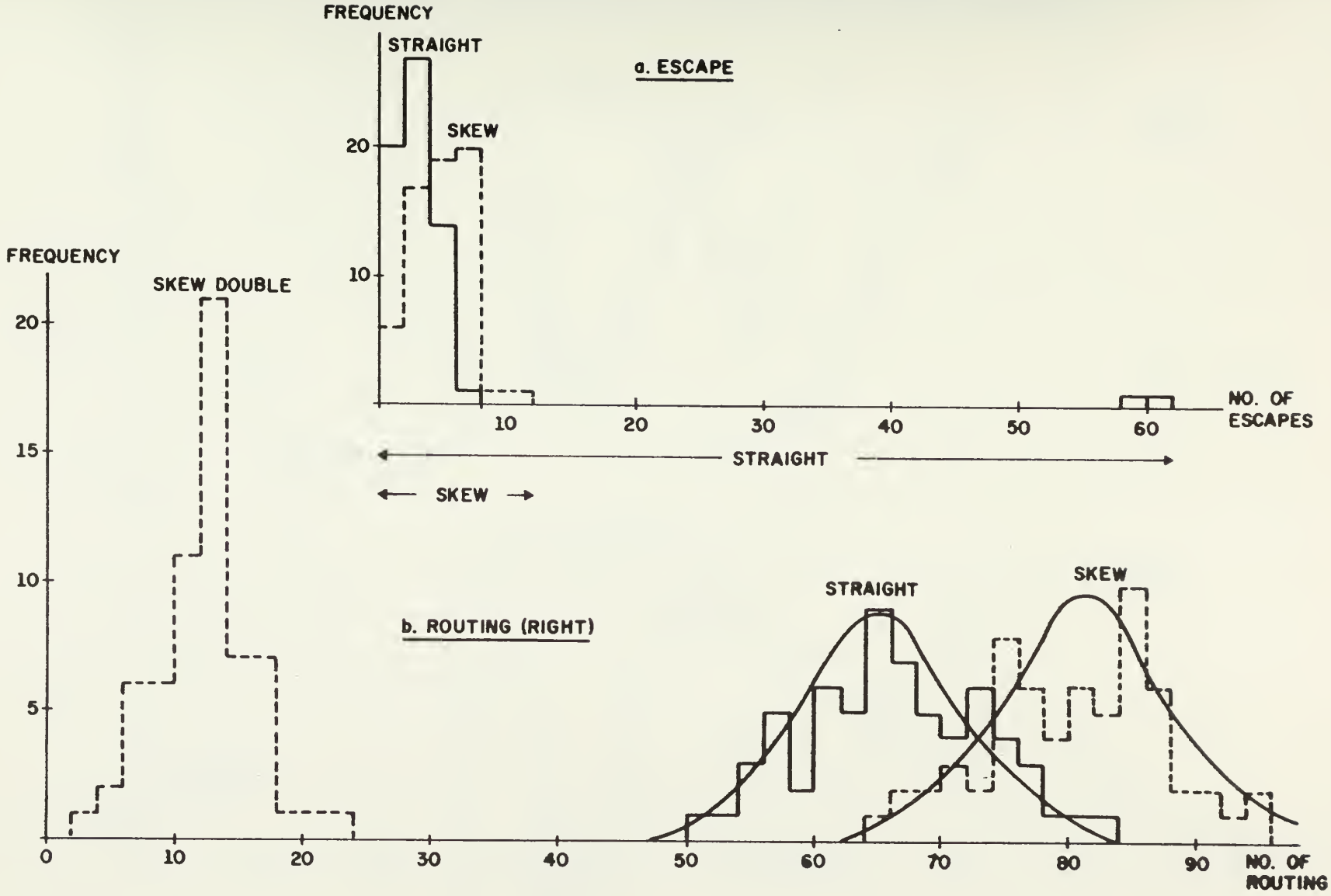


Figure 5.4 Straight Storage Vs Skewed Storage (64x8 Cells)
σ=.4, Vel.=1, Δt=1, and Δx=1, 64x8x40 particles

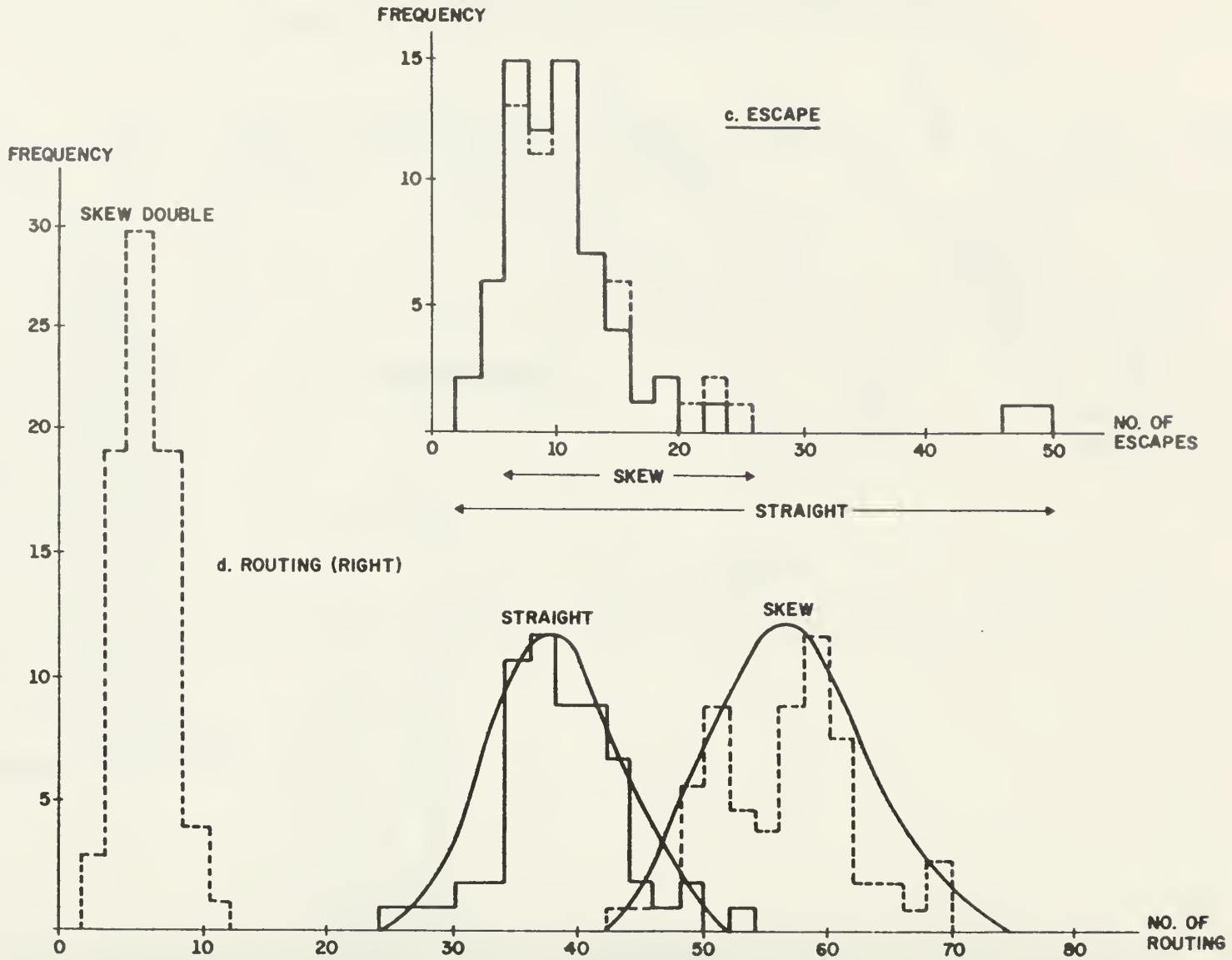


Figure 5.4 Straight Storage Vs Skewed Storage (64x8 Cells) (continued)
σ=.2, Vel.=.5, Δt=1, and Δx=1, 64x8x40 particles

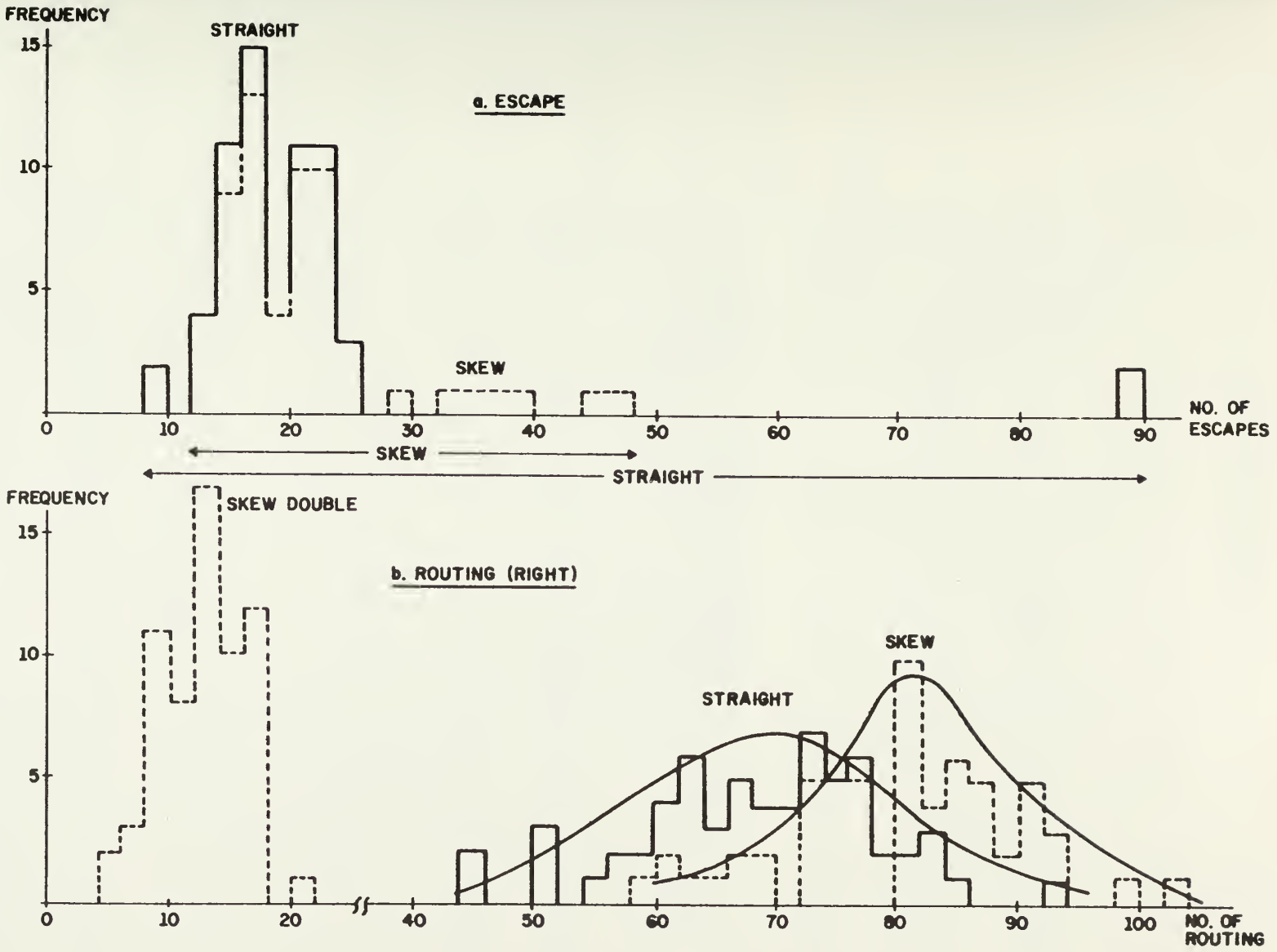


Figure 5.5 Straight Storage Vs Skewed Storage (64x64 Cells)
σ=.4, Vel.=1, Δt=1, and Δx=1, 64x64x5 particles

	0	1–21	22–43	44–63
PKL		CNB	XCO	YCO
		MU	COS PHI	SIN PHI
	$\xi\Delta t$		ENERGY	
	WEIGHT		τ	

	0–15	16–31	32–47	48–63
PNT	POINTER 1	POINTER 2	POINTER 3	POINTER 4

Figure 5.6 An Alternative Format for
the Particle Table (not used in the test program)

LIST OF REFERENCES

[1] Beckmann, P. A History of π, Golem Press, Boulder, Colorado, 1970.

[2] Chandrasekhar, S. Radiative Transfer, Dover Publication Inc., New York, 1960.

[3] Clark, M. Jr., and Hansen, K. Numerical Methods of Reactor Analysis, Academic Press, New York, 1964.

[4] Fleck, J. A. Jr. "The Calculation of Nonlinear Radiation Transport by a Monte Carlo Method", Methods in Computational Physics, vol. 1, Academic Press, New York, 1963.

[5] Greenberger, M. "An Apriori Determination of Serial Correlation in Computer Generated Random Numbers", Mathematical Computations, 15, 1961, pp. 383-389.

[6] Knuth, D. The Art of Computer Programming, vol. 2, Addison-Wesley Publishing Co., Reading Massachusetts, 1969.

[7] Kourganov, V. Basic Methods in Transfer Problems, Dover Publication Inc., New York, 1963.

[8] Lawrie, D. H. "A List Processing Language for ILLIAC IV", ILLIAC IV Document No. 322, Department of Computer Science, University of Illinois, April 1, 1969.

[9] MacLaren, M.D., and Marsaglia, G. "Uniform Random Number Generators", Journal of the Association of Computing Machinery, 12, 1965, pp. 83-88.

[10] Shrieder, Y. A. The Monte Carlo Method, Pergamon Press, New York, 1966.

[11] Slotnick, D. L. "The Fastest Computer", Scientific American, 1971, pp. 76-87.

[12] von Neumann, J. and Richtmyer, R. "Statistical Methods in Neutron Diffusion", J. von Neumann Collected Works, vol. 5, The MacMillian Company, New York, 1963.

[13] von Neumann, J. "Various Techniques Used in Connection with Random Digits", J. von Neumann Collected Works, vol. 5, The MacMillian Company, 1963.

[14] Winje, G. "Random Number Generators for ILLIAC IV", ILLIAC IV Document No. 179, Department of Computer Science, University of Illinois, June 20, 1969.

[15] Zerby, C. "A Monte Carlo Calculation of the Response of Gamma Ray Scintillation Counters", Methods in Computational Physics, vol. 1, Academic Press, New York, 1963.

[16] Abramowitz, M. and Stegun, I. A. Handbook of Mathematical Functions, Dover Publication Inc., New York, 1965.

[17] Grossman, G. "SSK/SSK Reference Manual", ILLIAC IV Document No. 178, Department of Computer Science, University of Illinois, June 3, 1969.

APPENDIX 1

As is discussed in Chapter 3, it is interesting to investigate the probability density functions of the maximum length of chain-linked tables among PEs. Let the problem be stated as follows: There are N PEs and each PE has a table of M entries each of which is selected with the probability $\sigma(0<\sigma<1)$ so that the average length of the chains is σM. What is the distribution of the maximum length of the chains? Assume that M is a large number and σ is neither too small nor too close to 1.

Theorem. If X_1, X_2 have the distribution functions $F_1(X)$, $F_2(X)$ respectively, the distribution function of the stochastic variable $X = \max(X_1, X_2)$ is $F(X) = F_1(X)F_2(X)$.

Proof. Let $P_1(X)$, $P_2(X)$ and $P(X)$ denote the probability density functions of X_1, X_2, and X, respectively. Then,

$$P(X)dX = P_1(X)dx \int^X P_2(X')dX' + P_2(X)dX \int^X P_1(X')dX'$$

$$= P_1(X)F_2(X)dX + P_2(X)F_1(X)dx$$

$$= \frac{d}{dX}(F_1(X)F_2(X))dX$$

$$\therefore F(X) = F_1(X)F_2(X). \qquad \text{Q.E.D.}$$

It is easy to derive the formula inductively when there are N independent variables. That is, $X = \text{Max}(X_1, X_2, \ldots X_n)$ has the distribution function $F(X) = F_1(X)F_2(X) \ldots F_n(X)$ where $F_1(X)$ is the distribution function of the ith variable.

Going back to the original problem, if the given assumptions are valid the distribution functions of the length of the chain in each PE can be approximated by the normal distribution:

$$\Phi(X_i) = \frac{1}{\sqrt{2\pi M\sigma\ (1-\sigma)}} \exp \int_{-\infty}^{X_i} \exp\ \{-\frac{(t-M\sigma)^2}{2M\sigma(1-\sigma)}\}\ dt \quad (i = 1, 2, \ldots N)$$

Therefore the distribution function of the maximum length $\Phi(X)$ is:

$$\Phi(X) = \{\Phi(X)\}^N \text{ or the probability density } P(X) \text{ is:}$$

$$P(X) = N\ \{\Phi(X)\}^{N-1}\ \frac{1}{\sqrt{2\pi M\sigma(1-\sigma)}}\ \exp\ \{-\frac{(t-M\sigma)^2}{2M\sigma(1-\sigma)}\}$$

Figure A1.1 shows P(X) for N(0,1) when N = 64. P(X) has the maximum value at X = 2.1. This was obtained by plotting values calculated from the mathematical table. (Ref. [16])

(Example) N = 64 (ILLIAC IV)

M = 100

σ = .3

$0 \leq Z \leq 100$

$\overline{Z}$ = 30

normalization of the variable

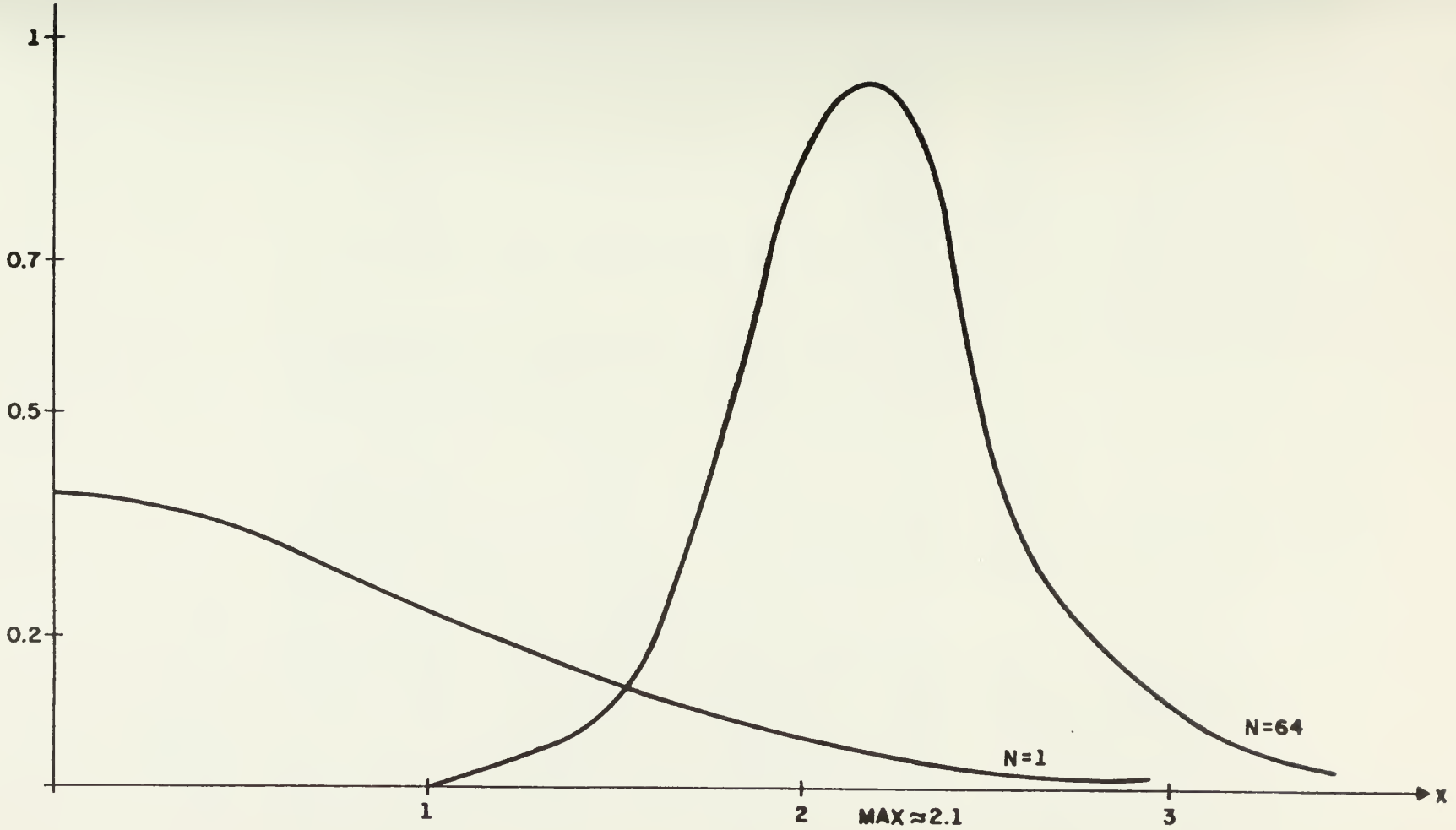
$$X = \frac{Z-30}{\sqrt{0.3x0.7x100}} = \frac{Z-30}{4.6}$$

X(Z = 100) = 15.2

X(Z = 0) = 16.5

Z(peak) = 30 + 2.1 x 4.6 = 39.7

Therefore the expectation value of the maximum length of the chain is 40.



$N=1$: NORMAL DISTRIBUTION $N(0,1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$

N=64: THE MAXIMUM VALUE OF 64 SAMPLES FROM N (0, 1)

Figure A.1 The Probability Density Function of the Maximum Value of 64 Samples from Normal Distribution

APPENDIX 2

$P_n(\ell)$ in section 5.2 is derived in the following way:

Suppose n-1 scatterings have happened in [0,x] and the nth scattering happens in [x, x+dx]. Then the probability that n scatterings happen anywhere in $[0,\ell]$ is

$$P_n(\ell) = \int_0^{\ell} \underbrace{P_{n-1}(x)}_{\text{n-1 scattering}} \cdot \underbrace{\sigma dx}_{\text{nth scattering}} \underbrace{Po(\ell-x)}_{\text{no scattering}}$$

$$Po(\ell) = \int_0^{\ell} \frac{1}{\sigma} e^{-\sigma x} dx = e^{-\sigma\ell}$$

Laplace transformation yields

$$q_n(s) = \int_0^{\infty} e^{-s\ell} P_n(\ell) d\ell$$

$$= \sigma q_{n-1}(s)\, q_o(s)$$

$$= \sigma^n q_o(s)^{n+1}$$

$$q_o(s) = \int_0^{\ell} e^{-s\ell-\sigma\ell} d\ell = \frac{1}{s+\sigma}$$

$$\therefore q_n(s) = \frac{\sigma^n}{(s+\sigma)^{n+1}}$$

Inverse Laplace transformation yields

$$P_n(\ell) = \frac{\sigma^n}{2\pi i} \int_{C-i\infty}^{C+i\infty} \frac{e^{s\ell}}{(\sigma + s)^{n+1}}\, ds \qquad (C>0 \text{ real})$$

$$= \sigma^n \text{ Res.}(s = -\sigma)$$

$$= \frac{(\sigma\ell)^n}{n!} e^{-\sigma\ell}$$

Q.E.D.

APPENDIX 3

```
                    CONGRATULATIONS. YOU ARE THE LUCKY NEW OWNER OF A LISTING FROM THE

                                I L L I A C  I V  G L Y P N I R  C O M P I L E R

                                        ( MARCH VERSION 2.5.01 )

                                    TUESDAY  AUGUST 10, 1971.   14:43

                                        ASK GOD: FILE1 MONTE/CARLO.

     BEGIN                                                                          •      1
       FILE LINE((1 WORDS))                                                         •      2
       FILE CARD((64 * 1WORDS))                                                     •      3
%                                                                                   •      4
%                                                                                   •      5
                %                                                 Y                 •      6
                %                                                 •                 •      7
                %                                                 Y                 •      8
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%•      9
%%      DECLARATION OF GLOBAL VARIABLES                                           •      10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%•      11
%                                                                                   •      12
%                                                                                   •      13
       REAL        THX         •REAL    LENGTH OF TIME STEP                       •      14
                    TWOPI      •REAL PI*2.0.....                                  •      15
                   STP         •REAL PROBABILITY THAT IS CONSTANT FOR RADIATION   •      16
                   AN          •REAL MULTIPLICATIVE CONSTANT FOR RANDOM NUMBER GENERA•     17
                               %F    10.7                                         •      18
                   CONST       •REAL ADDITIVE CONSTANT FOR RANDOM NUMBER GENERATOR. •     19
                   M17         •REAL MULTIPLIER THAT IS CONSTANT(USUALLY ABBREVIATED AS "M")•  20
                               ;                                                  •      21
       CINT        I,J,K,L,M   •INTEGERS                                          •      22
                   TIME        •INTEGER OF TIME CYCLE                             •      23
                   TNP         •INTEGER NUMBER OF PARTICLE TABLE                  •      24
                               ;                                                  •      25
       PINT        VCTR        •INTEGER OF VACANCY IN PARTICLE TABLE              •      26
                   SCTR        •INTEGER THAT PARTICLES TO BE SCATTERED            •      27
                   RCTR        •INTEGER OF PARTICLES TO BE ROUTED TO RIGHT        •      28
                   LCTR        •INTEGER OF PARTICLES TO BE ROUTED TO LEFT         •      29
                   VNXT        •INTEGER POINTER (OUT IN GLYPNIR SENSE) TO INDICATE •     30
                               INDEX OF CURRENT VACANCY                           •      31
                   SNXT        •INTEGER POINTER TO INDICATE A PARTICLE TO BE SCATTERED •  32
                               INDEX                                              •      33
                   RNXT        •INTEGER POINTER TO INDICATE A PARTILE TO BE       •      34
                   LNXT        •INTEGER POINTER TO INDICATE A PARTILE TO BE ROUTED LEFT •  35
                               INDEXED RIGHT NEXT                                 •      36
                   CNM         •INTEGER NUMBER OF A PARTICLE                      •      37
                   DPTH        •INTEGER(0:100) ARRAY                              •      38
                   RGLM        •INTEGER I MACHINE NUMBER GENERATOR                •      39
                   PCTR        •INTEGER I  PARTICLEASSIGN                         •      40
       PREAL        VCU        •REAL RELATIVE COORDINATE=X DIRECTION              •      41
                    VCV        •REAL RELATIVE COORDINATE=Y DIRECTION              •      42
                   DMP         •REAL DISTANCE TO CELL BOUNDARY                    •      43
                   TH          •REAL INTERIOR COSINE (X=DIRECTION=1)              •      44
                   SNF         •REAL VALUE OF AZIMUTHAL ANGLE                     •      45
                   CSF         •REAL COSINE OF AZIMUTHAL ANGLE                    •      46
                               REAL(POSITIVE X=DIRECTION=90 DEGREE)               •      47
```

```
                    TM        ,%%&RESIDUAL TIME AS A FRACTION OF TMX                    *      48
                    WT        ,%%&WEIGHTING FACTOR TO BE USED IN FREQUENCY              *      49
                              %%&DEPENDENT CASE(WT*ENR IS THE ACTUAL ENERGY             *      50
                              %%&CARRIED BY A PARTICLE).BECOMES ZERO AFTER              *      51
                              %%&ESCAPE OR ABSORPTION                                   *      52
                    ENR       ,%%&ENERGY PER UNIT WEIGHT                                *      53
                    SVD       ,%%&SURVIVAL DISTANCE (OR SO CALLED PATH LENGTH)          *      54
                    VEL       ,%%&VIRTUAL VELOCITY (PARTICLE DOES NOT GO AT             *      55
                              %%&THE VELOCITY OF LIGHT DUE TO SCALE ADJUSTMENT          *      56
                    ABC       ,%%&ABSORPTION COEFFICIENT                                *      57
                    SCC       ,%%&SCATTERING COEFFICIENT                                *      58
                    DNS       ,%%&DENSITY OF MEDIUM IN WHICH A PARTICLE IS              *      59
                              %%&TRAVELLING.(DEPENDS ON CELL NUMBER)                    *      60
                    DUMMY     ,%% DUMMY VARIABLE FOR RN)X (PREAL)                       *      61
                    SGM       ,%%& TOTAL CROSS SECTION OF INTERACTION                   *      62
                    DLTS      ,%%& LENGTH FOR A PARTICLE TO TRAVEL IN DELTA T           *      63
                    RN        ,%%& CURRENT RANDOM NUMBER                                *      64
                    ECTR      ,%%& ESCAPE COUNTER                                       *      65
                    ACTR      ,%%& ABSORPTION COUNTER                                   *      66
                    DGM       ,%%& DENSITY(TOTAL CROSS SECTION                          *      67
                              %%&(DIMENSION IS SAME AS SIGT)                            *      68
                    SPH       ,%%&SPECIFIC HEAT OF THE MEDIUM                           *      69
                     SUM      ;%%& USED FOR SUMMING UP SOME OTHER VARIABLES             *      70
          PCPOINT   PKL       ,%%&PARTICLE TABLE                                        *      71
                    DENS      ,%%&DENSITY TABLE                                         *      72
                    ENERGY    ,%%&ENERGY     TABLE                                      *      73
                    CHECK     ;%%&USED TO LABEL ROUTING,ESCAPE,INTERACTION ETC.         *      74
                              %%& X5:BOUNDARY CROSSING, X6:INTERACTION,                 *      75
                              %%& X7:ESCAPE, X8: ROUTING(= OR =1)                       *      76
          REAL FIELD  X0[0:FULL]      ,                                                 *      77
                      X1[0:INNER]     ,                                                 *      78
                      X2[0:OUTER]     ;                                                 *      79
          INTEGER FIELD                                                                 *      80
                       X3[0:0:16),                                                      *      81
                       X4[0:16:16),                                                     *      82
                       X5[0:32:16),                                                     *      83
                       X6[0:48:16),                                                     *      84
                       X7[0:0:8),                                                       *      85
                       X8[0:8:8));                                                      *      86
          ALPHA FIELD   X[0:0:64));                                                     *      87
$                                                                                       *      88
$                                                                                       *      89
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*      90
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*      91
$$        LIST OF SUBROUTINES                                                           *      92
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*      93
                $                                                                     *      94
                $                                                                     *      95
                $                                                                     *      96
  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*      97
     PREAL SUBROUTINE LN AS RGA(PREAL X AS RGA);                                        *      98
  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*      99
          BEGIN                                                                         *     100
$               INCINS      100
$$              TAPE2=GLYPNIR/GLN    SERIAL;
          END;%%%NATURAL LOG%%%                                                         *     101
  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*     102
```

```
                                                                    10
      PREAL SUBROUTINE EXP AS RGA(PREAL X AS RGA);                                  *   103
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   104
        BEGIN                                                                       *   105
%             INCINS     100
%             ASKRATCH=20
%%            TAPE2=GLYPNIR/GEXP   SERIAL;
        END;%%%EXPONENTIAL%%%                                                       *   106
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   107
                                                                    10%
      PREAL SUBROUTINE SQRT AS RGA(PREAL X AS RGA);                                 *   108
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   109
        BEGIN                                                                       *   110
%             INCINS     100
%             ASKRATCH=20
%%            TAPE2=GLYPNIR/GSQRT   SERIAL;
        END;%%%SQUARE ROOT%%%                                                       *   111
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   112
       PREAL SUBROUTINE SIN AS RGA(PREAL X AS RGA);                                 *   113
                                                                    11-
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   114
        BEGIN                                                                       *   115
%             INCINS     100
%             ASKRATCH=20
%%            TAPE2=GLYPNIR/GSIN   SERIAL;
        END;%%%SINE%%%                                                              *   116
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   117
     PREAL SUBROUTINE COS AS RGA(PREAL X AS RGA);                                   *   118
                                                                    11%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   119
        BEGIN                                                                       *   120
%             INCINS     100
%             ASKRATCH=20
%%            TAPE2=GLYPNIR/GCOS   SERIAL;
        END;%%%COSINE%%%                                                            *   121
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   122
                                                                    12-
       SUBROUTINE CHAIN(PINT OUT PCTR,PINT OUT PNTR,CINT Q,PINT                     *   123
                        PLOC);                                                      *   124
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   125
%          THIS IS ANEW VERSION,DUE TO THE NECCESITY OF USING TWO POINTERS,        *   126
%       THIS SUBROUTINE HAS BEEN MADE MORE FLEXIBLY,FOR ROUTING,                   *   127
%     Q=1,FOR VACANCY AND SCATTERING J=0.                                          *   128
            BEGIN                                                                   *   129
                PCTR:=PCTR+1;                                                       *   130
                PKL.[PLOC+Q]X41:=PNTR;                                              *   131
                PNTR:=PLOC+Q;                                                       *   132
              END; %% CHAIN %%                                                      *   133
                                                                    129
%                                                                                   *   134
%                                                                                   *   135
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   136
     PREAL SUBROUTINE  CVTRL(PINT A AS RGA);                                        *   137
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   138
%    HOW TO USE THIS ROUTINE. SPECIFY AS FOLLOWS;                                   *   139
%          SOME VARIABLE:=CVTRL(PNT.[I]X1);                                         *   140
%    THIS IS TO CONVERT 16 BIT INTEGER INTO 64 BIT REAL.                            *   141
```

```
          BEGIN                                                                 *   142
            PREAL R AS R=A,SS AS R=S;                                           *   143
$*              SHAL    33;             % SHIFT MANTISSA LEFT
$*              LDS     =A;             % RGS STILL INTEGER TYPE
                SS.[1:15]:=40001(A);    %EXPONENT PART                          *   144
$*              LDA     =S;
$*              NORM;
                CVTRL:=R;                                                       *   145
          END; % CVTRL **                                                       *   146
                                                                  142
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   147
          PREAL SUBROUTINE  RNDX(CINT IX);                                      *   148
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   149
%%              RANDOM NUMBER GENERATER BY MIXED CONGRUENTIAL METHOD.           *   150
%%              IF IX=0 THEN INTEGER TYPE RANDOM NUMBERS ARE IN "RGLH",         *   151
%%              IN THIS CASE RNDX IS ASSIGNED TO A DUMMY VARIABLE.              *   152
%%              IF IX=1 THEN REAL TYPE RANDOM NUMBERS ARE IN "RNDX".            *   153
%%                                                                              *   154
%%                                                                              *   155
          BEGIN                                                                 *   156
              PINT AA AS RGA,BB AS RGB,SS AS RGS;                               *   157
              PREAL AR AS RGA,SR AS RGS;                                        *   158
                  AA:=AA;   %MULTIPLIER                                         *   159
                  BB:=RGLH; %OLD RANDOM NUMBER                                  *   160
$*                LLM   BB;
                  AA:=CONST;                                                    *   161
$*                ADM   =A;
                  RGL:=AA; %NEW RANDOM NUMBER (INTEGER)                         *   162
                IF IX EQL 1 THEN                                                *   163
                  BEGIN                                                         *   164
$*                  LDS   =A;
                    SR.[1:15]:=4000(A);  %CONVERSION TO REAL TYPE               *   165
$*                  LDA   =S;
$*                  NORM;
                    RNDX:=AR;                                                   *   166
                  END;                                                          *   167
                                                                  164
            END;  %RANDOM %                                                     *   168
                                                                  156
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   169
        SUBROUTINE DATALOAD;                                                    *   170
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   171
          BEGIN                                                                 *   172
          PREAL     TEMP;                                                       *   173
                  SYSREAD(CARD,RGL);      %%READ INITIAL RANDOM NUMBER          *   174
              SYSREAD(CARD,ABC,SCC,THX,STP);                                    *   175
                  TAUPI:=6.28318530717958;                                      *   176
              LOOP I:=0,1,7 DO                                                  *   177
                BEGIN                                                           *   178
                  SYSREAD(CARD,TEMP);                                           *   179
                  DENS.[I]X1:=TEMP; %DENSITY                                    *   180
                  SYSREAD(CARD,TEMP);                                           *   181
                  ENERGY.[I]X1:=TEMP;  % TEMPERATURE                            *   182
                  SYSREAD(CARD,TEMP);                                           *   183
                  DENS.[I]X2:=TEMP;   %SPECIFIC HEAT                            *   184
                  ENERGY.[I]X2:=0; %% TALLY CLEARED.                            *   185
                END;                                                            *   186
```

```
                                                                    174
        END; %DATALOAD                                                      *   187
                                                                    173
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   188
    PINT SUBROUTINE MOD(PINT A);                                            *   189
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   190
%   THIS SUBROUTINE IS USED TO CALCULATE THE "DEPTH" OF TABLE ENTRIES.      *   191
%            MOD=A MOD B.   (FOR 64*B  CELLS)                               *   192
%   IN CASE OF 64*64 CELLS, CHANGE THE PARTIAL WORD DESIGNATOR TO [5:11]    *   193
        BEGIN                                                               *   194
          MOD:=A.[61:3];                                                    *   195
        END %% MOD %%                                                       *   196
                                                                    194
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   197
    SUBROUTINE BIRTH;                                                       *   198
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   199
%   THIS SUBROUTINE IS FOR CREATION OF ALL THE PARTICLES. ALL THE           *   200
%                                                                           *   201
%                                                                           *   202
%   NULLS ARE CONNECTED USING SNXT.SO.LINK MUST BE SWITCHED FROM VNXT       *   203
%   TO SNXT. FIRST HALF OF THIS SUBROUTINE IS FOR T                         *   204
    BEGIN                                                                   *   205
        PINT Q,TMP;                                                         *   206
        PREAL THETA;                                                        *   207
            SCTR:=0;                                                        *   208
%           ONLY 2/3 OF VACANT TABLES ARE TO BE FILLED WITH NEW PARTICLES.  *   209
            FOR Q:=1 STEP 1 UNTIL VCTR*.6 DO                                *   210
              BEGIN                                                         *   211
                TMP:=VNXT;                                                  *   212
                VNXT:=PKL.[TMP]X4;                                          *   213
                PKL.[TMP]X4:=SNXT;                                          *   214
                SNXT:=TMP;                                                  *   215
                VCTR:=VCTR-1;                                               *   216
                SCTR:=SCTR+1;                                               *   217
              END;  %% FOR  %%                                              *   218
                                                                    211
            SIMWRITE(LINE,"BIRTH CALLED",## OF PARTICLES=",SCTR);           *   219
            FOR Q:=1 STEP 1 UNTIL SCTR DO                                   *   220
              BEGIN                                                         *   221
%   THIS BLOCK      IS TO CREATE A NEW PARTICLE IN EACH PE,                 *   222
%   CELL NUMBER HAS BEEN DETERMINE, THEREFORE,RELATIVE COORDINATES          *   223
%   XCO,YCO,DIRECTION COSINE MU, AZIM, CSF AND SNF, TIME OF CREATION        *   224
%   AS A FRACTION OF DELTA T.TM, SURVIVAL DISTANCE SVD ARE DETERMINED       *   225
%   HERE                                                                    *   226
                PKL.[SNXT]X3:=PEN*8; %FOR UNIFORM INCIDENT BEAM             *   227
                DPTH:=MOD(PKL.[SNXT]X3);                                    *   228
                PKL.[SNXT]X5:=16;   %FOR TESTING                            *   229
            %%  DUMMY:=RNDX(0);                                             *   230
            %%  PKL.[SNXT]X5:=RGLR.[16:15];   %XCO                          *   231
                DUMMY:=RNDX(0);                                             *   232
                PKL.[SNXT]X6:=RGLR.[16:15];   %YCO                          *   233
                SNXT:=SNXT+1;                                               *   234
                DUMMY:=RNDX(0);                                             *   235
                PKL.[SNXT]X5:=RGLR.[16:15];   %POSITIVE DIRECTIONS          *   236
            %%  PKL.[SNXT]X5:=2*RGLR.[16:15]-1;    % ALL DIRECTIONS         *   237
                THETA:= TWOPI*RNDX(1);                                      *   238
                PKL.[SNXT]X3:=SIN(THETA)*16384;                             *   239
```

```
                PKL.[SNXT]X6:=COS(THETA)*1634;                                  *  240
                PKL.[SNXT+1]X1:=RNDX(1)*TMX;          %TIME OF CREATION         *  241
                PKL.[SNXT+2]X2:=-LN(1-RNDX(1));                                 *  242
                SNXT:=PKL.[SNXT+1]X4;                                           *  243
              END;                                                              *  244
                                                                    221
                SCTR:=0;                                                        *  245
        END; %BIRTH                                                             *  246
                                                                    205
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*  247
          SUBROUTINE TALLY;                                                     *  248
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*  249
          BEGIN                                                                 *  250
%           SUM UP THE PARTICLES IN EACH CELL.                                  *  251
%                                                                               *  252
            LOOP TNR:=1,4,76 DO                                                 *  253
              BEGIN                                                             *  254
                DPTH:=MOD(PKL.[TNR]X3);                                         *  255
                ENERGY.[DPTH]X1:=ENERGY.[DPTH]X1+PKL.[TNR+2]X2*                 *  256
                                 PKL.[TNR+3]X1;                                 *  257
              END;                                                              *  258
                                                                    254
          END; %% TALLY %%                                                      *  259
                                                                    250
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*  260
          SUBROUTINE  ENERGYBALANCE;                                            *  261
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  *  262
          BEGIN                                                                 *  263
%                                                                               *  264
%           THIS SUBROUTINE IS TO CALCULATE THE NEW ENERGY AND THE              *  265
%           RADIATION IN THE NEXT STEP. THIS SUBROUTINE SHOULD BE               *  266
%           CALLED AFTER PROCESSING PARTICLES, AND IS TO BE FOLLOWED            *  267
%           BY "PARTICLEASSIGN".                                                *  268
%                                                                               *  269
%           GLOBAL VARIABLES ARE                                                *  270
%           PREAL  :SPH(SPECIFIC HEAT  IN CGS UNIT.)                            *  271
%                  STP (STEPHAN-BOLTZMANN S CONSTANT FOR BLACK BODY             *  272
%                       RADIATION, IN CGS UNIT.)                                *  273
%                  SUM (TO TAKE SUM OF RATIATIVE ENERGY.)                       *  274
%           CINT   :TMX (LENGTH OF TIME STEP.)                                  *  275
%           POINTER:                                                            *  276
%                  ENERGY   (STORED ENERGY IN EACH CELL)                        *  277
%                  DENS     (DENSITY OF EACH CELL)                              *  278
%                  CFAV  (FREQUENCY-AVERAGED ABSORPTION COEFFICIENT)            *  279
%                                                                               *  280
%                                                                               *  281
            CINT   DPTH;                                                        *  282
           PREAL  XY,XZ;                                                        *  283
           CREAL  CFAV;                                                         *  284
            CFAV:=ABC;                                                          *  285
            SUM:=0;                                                             *  286
            LOOP DPTH:=0,1,7 DO                                                 *  287
              BEGIN                                                             *  288
                XZ:=ENERGY.[DPTH]X1;      % STORED ENERGY READ OUT             *  289
                SPH:=DENS.[DPTH]X2;       % SPECIFIC HEAT                      *  290
                XY:=XZ/SPH;               % TEMPERATURE                        *  291
                XY:=XY*XY;                % TEMP **4                           *  292
```

```
                    XY:=XY*XY;                                                    *    293
                    DGM:=CFAV*STP*THY*DENS.[DPTH]X1;                              *    294
                    XY:=DGM*XY;                                                   *    295
                    XZ:=XZ+XY*DGM*ENERGY.[DPTH]X2;                                *    296
                    ENERGY.[DPTH]X1:=XZ;  %% STORED ENERGY OF THIS CELL.          *    297
                    ENERGY.[DPTH]X2:=XY;  %% RADIATIVE ENERGY                     *    298
                  END;%% LOOP %%                                                  *    299
                    SUM:=SUM+XY;                                                  *    300
                                                                          284
            END; %% ENERGYBALANCE %%                                              *    301
                                                                          263
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    302
          SUBROUTINE  PARTICLEASSIGN;                                             *    303
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    304
            BEGIN                                                                 *    305
%                                                                                 *    306
%         THIS SUBROUTINE IS TO DISTRIBUTE PARTICLE TABLE TO CELLS.               *    307
%         ENERGY DISTRIBUTION IS NOT CONSIDERED.ENERGY IS CONSERVED.              *    308
%         CELL NUMBERS OF PARTICLES ARE ALSO ASSIGNED HERE. VACANCY               *    309
%                                                                                 *    310
%     CHAIN IS TRANSFERED TO SCATTER CHAIN TO CONNECT ALL NEW PARTICLES.          *    311
%                                                                                 *    312
%         GLOBAL VARIABLES ARE:                                                   *    313
%             PREAL:                                                              *    314
%                   SUM (SUM OF RADIATIVE ENERGY IN EACH PE)                      *    315
%             PINT:                                                               *    316
%                   VNXT     (VACANCY POINTER)                                    *    317
%                   VCTR     (VACANCY COUNTER)                                    *    318
%                   SNXT     (NEW PARTICLE POINTER)                               *    319
%                   SCTR     (NEW PARTICLE COUNTER)                               *    320
%                             PARTICLES)                                          *    321
%                   PCTR     (NUMBER OF TABLE ENTRIES AVAILABLE FOR NEW           *    322
%             POINTER:                                                            *    323
%                                                                                 *    324
%                   ENERGY   (STORED ENERGY TABLE).                               *    325
%                   PKL      (PARTICLE TABLE)                                     *    326
%                                                                                 *    327
%                                                                                 *    328
            PINT    J,K,TMP,PNO;                                                  *    329
            CINT  DPTH;   %(NN: GLOBALLY "DPTH IS PE INTEGER.)                    *    330
            PREAL   XY,XZ;                                                        *    331
            PCTR:=VCTR*.667; %ONLY 2/3 OF THEVACANCY IS                           *    332
                             %USED FOR NEW PARTICLE.                              *    333
                LOOP DPTH:=0,1,7 DO                                               *    334
                  BEGIN                                                           *    335
                  K:=ENERGY.[DPTH]X2/SUM*PCTR; %%K INDICATED TABLE                *    336
                                               %%ENTRIES ARE ASSIGNED             *    337
                                               %%TO THIS CELL.                    *    338
                    SIMWRITE(LINE,"K=",K);                                        *    339
                    FOR PNO:=1 STEP 1 UNTIL K DO                                  *    340
                      BEGIN                                                       *    341
                        PKL.[VNXT]X3:=PEN*8+DPTH;    %% CELL NUMBER            *    342
                        PKL.[VNXT+3]X1:=1; %%WEIGHTING FACTOR                     *    343
                        PKL.[VNXT+2]X2:=XZ/PCTR;                                  *    344
                                       %ENERGY IS ASSIGNED TO                     *    345
                                       %A PARTICLE IN MONO-                       *    346
                                       %CHROMATIC CASE. THAT                      *    347
```

```
                                        %IS THE WEIGHTING FACTOR                    *    348
                                        %WT IS PUT TO 1.                            *    349
                                        %IN NON-MONOCHROMATIC CASE                  *    350
                                        %ENERGY MUST BE SELECTED                    *    351
                                        %RANDOMLY FROM PLANCKIAN                    *    352
                                        %DISTRIBUTION AND                           *    353
                                        %THE WT MUST BE ADJUSTED                    *    354
                                        %SO THAT THE TOTAL ENERGY                   *    355
                                        %IS CONSERVED.                              *    356
                                SIM WRITE(LINE,"CELL NO.",PKL.[VNXT]X3,             *    357
                                          "ENR",PKL.[VNXT+2]X2);                    *    358
                                SCTR:=SCTR+1;                                       *    359
                                VCTR:=VCTR-1;                                       *    360
                                TMP:=VNXT;                                          *    361
                                VNXT:=PKL.[VNXT]X4;                                 *    362
                                PKL.[TMP]X4:=SNXT;                                  *    363
                                SNXT:=TMP;                                          *    364
                              END; %% OF A CELL %%                                  *    365
                                                                    341
                            END; %% OF DEPTH LOOP %%                                *    366
                                                                    335
                          SIMWRITE(LINE,"PARTICLEASSIGN CALLED");                   *    367
                        END; %% PARTICLE ASSIGN %%                                  *    368
                                                                    309
%***************************************************************************      *    369
    SUBROUTINE DSTBND    ;                                                        *    370
%***************************************************************************      *    371
%                                                                                *    372
%                                                                                *    373
%    THIS SUBROUTINE IS TO CALCULATE THE DISTANCE TO THE CELL                    *    374
%    BOUNDARY. THIS QUANTITY (DBR) AND THE SURVIVAL DISTANCE                     *    375
%    (SVD) DETERMINE WHERE AND WHEN THE INTERACTION (ABSORPTION,                 *    376
%    SCATTERING ETC.) IS GOING TO HAPPEN.                                        *    377
%                                                                                *    378
%                                                                                *    379
        BEGIN                                                                   *    380
          REAL TMP1,TMP2;                                                       *    381
               FOR ALL MU>0 DO                                                  *    382
                    DBR:=(1-XCO)/MU;                                            *    383
               FOR ALL MU<0 DO                                                  *    384
                    DBR:=-XCO/MU;                                               *    385
               FOR ALL MU=0 DO                                                  *    386
                    DBR:=100;                                                   *    387
               TMP1:=SQRT(1-MU*MU)*SNF;                                         *    388
               FOR ALL TMP1=0  DO                                               *    389
                    TMP2:=100;                                                  *    390
               FOR ALL TMP2  NE  0 DO                                           *    391
                 BEGIN                                                          *    392
                    FOR ALL  SNF<0 DO                                           *    393
                         TMP2:=-YCO/TMP1;                                       *    394
                    FOR ALL  SNF>0 DO                                           *    395
                         TMP2:=(1-YCO)/TMP1;                                    *    396
                 END;                                                           *    397
                                                                    392
               FOR ALL TMP2<DBR DO                                              *    398
                    DBR:=TMP2;                                                  *    399
        END; % DSTBND %                                                         *    400
```

```
                                                                        78
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*
    SUBROUTINE  ESCAPE;                                                     *
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*
%                                                                           *
%                                                                           *
%   THIS SUBROUTINE IS TO TREAT PARTICLES WHICH ESCAPES OUT OF THE          *
%   MEDIUM,TABLES FOR THOSE PARTICLES ARE DESTORYED AND THE                 *
%   COUNTER IS INCREMENTED BY WEIGHT.                                       *
%                                                                           *
%                                                                           *
        BEGIN                                                               *
                ECTR:=ECTR+WT;                                              *
                CHECK.X5:=4096;                                             *
                WT:=0;                                                      *
                CHAIN(VCTR,VNXT,TNB,0);                                     *
       END;   %%ESCAPE FROM THE MEDIUM                                      *
                                                                        411
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*
       SUBROUTINE      ABSORB;                                              *
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*
        BEGIN                                                               *
                ACTR:=ACTR+WT;        %ABSORPTION COUNTER INCREMENTED.      *
                WT:=0;                %TO IDENTIFY AN ABSORBED PARTICLE     *
                CHAIN(VCTR,VNXT,0,TNB);                                     *
             END;  %% ABSORB %%                                             *
                                                                        42
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*
      SUBROUTINE   TRANSLATE;                                               *
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*
        BEGIN                                                               *
%                                                                           *
%                                                                           *
%       THIS SUBROUTINE HAS TWO FUNCTIONS. ONE IS TO TRANSLATE              *
%       PARTICLES TO NEW POSIONS ACCORDING TO VELOCITY,DIRECTION            *
%       COSINE.  THE OTHER IS TO DITECT PARTICLES WHICH CROSS CELL          *
%       BOUNDARIES OR THOSE WHICH ESCAPE FROM THE MEDIUM AND                *
%       TO LABEL THEM USING CHECK.X8 (ROUTE CHECK) AND CHECK.X7 (           *
%       ESCAPE CHECK),RESPECTIVELY.                                         *
%                                                                           *
%       THIS SUBROUTINE IS TO BE IMBEDDED IN SUBROUTINE BRANCH.             *
%                                                                           *
%                                                                           *
%                                                                           *
%                                                                           *
          PREAL    Z;                                                       *
                XCO:=XCO+DLTS*MU;          %XCO INCREMENTED                 *
                YCO:=YCO+DLTS*SQRT(1-MU*MU)*SNF;   %YCO INCREMENTED         *
                TM:=TM-DLTS/VEL;           %TIME DECREMENTED                *
                SVD:=SVD-DLTS*UGM;     %SURVIVAL DISTANCE DECREMENTED       *
%                                                                           *
%                                                                           *
%                                                                           *
                Z:=XCO;                                                     *
                FOR ALL Z GEQ  .9999 DO                                     *
                  BEGIN                                                     *
                     XCO:=0;                                                *
```

```
                        CNBI=CNB+1;     %GOES TO RIGHT                          *    455
                        FOR ALL MOD(CNB)=0 DO                                   *    456
                             CHECK.X7I=1;    %ESCAPE                            *    457
                      END;  %%RIGHT%%                                           *    458
                                                                    453
%                                                                               *    459
%                                                                               *    460
                    FOR ALL Z LE) .0001 DO                                      *    461
                      BEGIN                                                     *    462
                        XCO;=1;                                                 *    463
                        CNBI=CNB-1;     %GOES TO LEFT                           *    464
                        FOR ALL MOD(CNB)=7 DO                                   *    465
                             CHECK.X7I=1;                                       *    466
                      END;   %%LEFT%%                                           *    467
                                                                    462
%                                                                               *    468
%           BOUNDARY CROSSING IN X-DIRECTION AND ESCAPING.                      *    469
%           NO ROUTING INVOLVED.                                                *    470
%                                                                               *    471
%                                                                               *    472
%           BOUNDARY CROSSING IN Y-DIRECTION. ROUTING INVOLVED.                 *    473
%           CARE HAS BEEN TAKEN FOR A DOUBLE ROUTING NOT TO                     *    474
%           OCCUR. IF A PARTICLE CROSSES A BOUNDARY DOWNWARD                    *    475
%           THEN UPWARD AGAIN AFTER SCATTERING,IT HAS                           *    476
%           TO BE ROUTED TWICE. THIS IS A LOSS OF TIME AND IT ALSO              *    477
%           CUTS OFF THE ROUTE LINK,BECAUSE RIGHT AND LEFT SHARES THE           *    478
%           SAME POINTER.                                                       *    479
                    DPTH:=MOD(CNB);                                             *    480
                    ZI=YCO;                                                     *    481
                    FOR ALL Z GE) .9999 DO                                      *    482
                      BEGIN                                                     *    483
                        YCO;=0;                                                 *    484
                        CNBI=CNB+8;     %GOES UPWARD                            *    485
                        CHECK.XBI=CHECK.XB+1;                                   *    486
                      END; %UP%                                                 *    487
                                                                    483
%                                                                               *    488
                    FOR ALL Z LE) .0001 DO                                      *    489
                      BEGIN                                                     *    490
                        YCO;=1;                                                 *    491
                        CNBI=CNB-8;     %GOES DOWNWARD                          *    492
                        CHECK.XBI=CHECK.XB-1;                                   *    493
                      END;  %DOWN%                                              *    494
                                                                    49-
%                                                                               *    495
%                                                                               *    496
%                                                                               *    497
%                                                                               *    498
                    FOR ALL CNB<0 OR CNB>511 DO                                 *    499
                      BEGIN                                                     *    500
                        CHECK.X7I=1;                                            *    501
                        CHECK.X5I=1024;                                         *    502
                      END;     %ESCAPE IN Y-DIRECTION                           *    503
                                                                    50O
           END; %%%TRANSLATE                                                    *    504
                                                                    42A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    505
```

```
        SUBROUTINE  BRANCH(PINT  TNB);
¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢¢
          BEGIN
¢
¢    THIS SUBROUTION IS TO DECIDE WHAT INTERACTION IS TO TAKF PLACE.
¢    AFTER THAT TRANSLATE IS CALLED, THEN ROUTE LINKING IS PERFORMED.
¢
¢
¢
¢
¢
¢
¢
¢
             PREAL    XY,XZ,XW;
             LABEL    L13;
                      CHECK.X3:=0;
           L13:       FOR ALL CHECK.X5=8192 DO
                        BEGIN
                           CHECK.X5:=4096;
                           DSTAND;
                           DLTS:=VEL*TM;
                           XY:=DRR-DLTS;
                           DGM:=[CHECK.X3]DENS.[DPTH]X1*SGM;
                           XZ:=SVD-DLTS*DGM;
                           XW:=DGM*DRR-SVD;
                           FOR ALL XY<0 AND XW LEQ 0 DO ¢CROSSING BOUNDARY
                             BEGIN
                                CHECK.X5:=8192;
                                DLTS:=DRR;
                             END;
                                                                              533
                           FOR ALL XY GEQ 0 AND XZ<0 OR XY<0 AND XW>0 DO
                             BEGIN   ¢INTERACTION
                                DLTS:=SVD/DGM;
                                CHECK.X6:=4096;
                             END;
                                                                              538
¢
¢                           THOSE ARE TO HAVE INTERACTION
¢
                           TRANSLATE;
¢
                           GO TO L13;
                       END;
                                                                              524
                      FOR ALL CHECK.X7=1 DO
                           ESCAPE;
                      FOR ALL CHECK.X3=1 DO     ¢RIGHT ROUTING
                           CHAIN(RCTR,RNXT,1,TNB);
                      FOR ALL CHECK.X3=-1 DO
                           CHAIN(LCTR,LNXT,1,TNB);
¢¢                   THOSE ARE FOR "ESCAPE"
                      CHECK.X8:=0;
                      CHECK.X5:=4096;
                     FOR ALL CHECK.X6=4096 AND WT>.01 DO ¢INTFRACTION TYPF
                       BEGIN
```

```
                        XY:=AHC/SGM;                                          *    560
                        RN:=RNDX(1);                                          *    561
                        FOR ALL RN LEQ XY DO                                  *    562
                             ABSORB;                                          *    563
                        FOR ALL RN>XY DO                                      *    564
                             CHAIN(SCTR,SNXT,0,TNB);  %SCATTERING             *    565
                   END;     %INTERACTION                                      *    566
                                                              550
%                                                                             *    567
          SIMWRITE(LINE,"BRANCH CALLED","ESCK=",CHECK,/7,                     *    568
                   "RTCK=",CHECK,X8);                                         *    569
     END;  %%BRANCH                                                          *    570
                                                              504
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    571
    SUBROUTINE STORE(PINT INB);                                               *    572
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    573
     BEGIN                                                                    *    574
%    TO STORE DATA ON A PARTICLE TO PARTICLE TABLE                            *    575
          PKL.(TNB)X3:=CNB;    %CELL NUMBER                                   *    576
          PKL.(TNB)X5:=XCO*16384;  %%COORDINATE                               *    577
          PKL.(TNB)X6:=YCO*16384;  %YCOORDINATE                               *    578
          PKL.(TNB+1)X5:=MU*16384; %DIRECTION COSINE                          *    579
          PKL.(TNB+1)X3:=SNF*16384;%SIN OF AZIM.ANGLE                         *    580
          PKL.(TNB+1)X6:=CSF*16384;%COS OF AZIM.ANGLE                         *    581
          FOR ALL CHECK,X8 NEQ 4096 AND WT NEQ 0 DO                           *    582
          PKL.(TNB+2)X1:=1; %%DELTA T.                                        *    583
          PKL.(TNB+3)X1:=WT;  %%EIGHT                                         *    584
          PKL.(TNB+3)X2:=SVD; %SURVIVAL DISTANCE                              *    585
     END; %STORE                                                              *    586
                                                              57;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    587
    SUBROUTINE ANGLE;                                                         *    588
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    589
%                                                                             *    590
%    DETERMINATION OF SCATTERING ANGLES. DIRECTION COSINE OBEYS               *    591
%    THE LAW OF THOMSON SCATTERING,I.E. PROB(MU)=3/8(1+MU*MU),                *    592
%    AZIMUTHAL ANGLE IS UNIFORM.                                              *    593
%                                                                             *    594
%                                                                             *    595
     BEGIN                                                                    *    596
      PREAL  XX,XY,XZ,XW,T1,T2,T3,THETA,KI;                                   *    597
       CINT  I;                                                               *    598
            %A RANDOM NUMBER WITH THE PROB. DENSITY GIVEN ABOVE               *    599
             XY:=.7;   %%INITIAL VALUE OF NEWTON-RAPHSON'S METHOD             *    600
             XW:=RNDX(1);                                                     *    601
             LOOP I:=1,1,3 DO    %NEWTON-RAPHSON ITERATION                    *    602
               BEGIN                                                          *    603
                  XZ:=XY*XY;                                                  *    604
                  XY:=(XY*XZ+.5-XW)*1.333333/(1+XZ);                          *    605
               END;                                                           *    606
                                                              603
             KI:=XY;   %SCATTERING ANGLE (DIRECTION COSINE)                   *    607
             THETA:=TWOPI*RNDX(1);     %AZIM.ANGLE                            *    608
             XW:=SQRT(1-KI*KI);        %SIN(KI)                               *    609
             XX:=SQRT(1-MU*MU);        %SIN(ORIGINAL DIRECTION)               *    610
             XZ:=XW*XX;                                                       *    611
%    CALCULATION OF NEW DIRECTION COSINE AND AZIMUTHAL ANGLE OF PARTICLES.    *    612
```

```
                FOR ALL XW NEQ 0 DO                                          *    613
                  BEGIN                                                      *    614
                    XY:=MU*KI=X/*COS(THETA);                                 *    615
                    T1:=SQRT(1-XY*XY);                                       *    616
                    FOR ALL T1 NEQ 0 DO     % NEW SET OF AZIM.               *    617
                      BEGIN                                                  *    618
                        T2:=XW*SIN(THETA)/T1;                                *    619
                        T3:=(MU*XY-KI)/(AX*T1);                              *    620
                        T1:=CSF;                                             *    621
                        CSF:=T2*SNF+T3*CSF;                                  *    622
                        SNF:=T3*SNF-T2*T1;                                   *    623
                      END;                                                   *    624
                                                              614
                  END;                                                       *    625
                                                              613
                FOR ALL MU=1 OR MU=-1 DO                                     *    626
                  BEGIN                                                      *    627
                    SNF:=SIN(THETA);                                         *    628
                    CSF:=COS(THETA);                                         *    629
                  END;                                                       *    630
                                                              627
                MU:=XY;                                                      *    631
                XW:=RNDX(1);                                                 *    632
                FOR ALL XW>.5 DO                                             *    633
                  MU:=-MU;                                                   *    634
        END;    % NEW ANGLES                                                 *    635
                                                              596
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    636
      SUBROUTINE  SCATTER;                                                   *    637
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    638
        BEGIN                                                                *    639
%                                                                            *    640
%                                                                            *    641
%     THOMSON SCATTERING  NEW DIRECTION COS, AZIM. ANGLES, TM AND SURVIVAL   *    642
%     DISTANCE ARE SELECTED                                                  *    643
%                                                                            *    644
%                                                                            *    645
          PINT  TNB;                                                         *    646
          PREAL THETA;                                                       *    647
          LABEL LS;                                                          *    648
               SIMWRITE(LINE,"SCATTERING CALLED");                           *    649
               SIMWRITE(LINE,"SCATTERED PARTICLE=",SNXT);                    *    650
          LS:  FOR ALL SCTR>0 DO                                             *    651
                 BEGIN                                                       *    652
                   CNB:=PKL.[SNXT]X3;  %CELL NUMBER                          *    653
                   DPTH:=MOD(CNB);                                           *    654
                   XCO:=CVTRL(PKL.[SNXT]X5);                                 *    655
                   YCO:=CVTRL(PKL.[SNXT]X6);                                 *    656
                   ANGLE;                                                    *    657
                   TMP:=PKL.[SNXT+2]X1;                                      *    658
                   DGM:=DENS.[DPTH]X1*SGM;                                   *    659
                   ENR:=PKL.[SNXT+2]X2;                                      *    660
                   WT:=PKL.[SNXT+3]X1;                                       *    661
                   SVD:=-LN(1-RNDX(1));                                      *    662
                   FOR ALL WT GEQ .01 DO                                     *    663
                        CHECK.A5:=#192;      %ENABLE PE:S IN "BRANCH"         *    664
                   BRANCH(SNXT);                                             *    665
```

```
                    PKL.[SNXT+2]X1:=TM;                                    *   466
                    CHECK.X6:=4095;                                        *   467
                    STORE(SNXT);                                           *   468
                    SNXT:=PKL.[SNXT]X4;                                    *   469
                    SCTR:=SCTR-1;                                          *   470
                    GO TO LS;                                              *   471
                  END;                                                     *   472
                                                                  65>
        END;  %SCATTERING                                                  *   473
                                                                  439
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   474
      SUBROUTINE ROUTE(PINT OUT CNTR,PINT OUT PNTR,CINT V);                *   475
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   476
%     THIS SUBROUTINE ROUTES PARTICLES TO THEIR NEIGHBORS.EITHER RIGHT      *   477
%     OR LEFT ROUTE CAN BE SPECIFIED BY SETTING K=1 OR K=3,RESPECTIVE       *   478
%           LY.  PNTR IS RNXT OR LNXT, WHEREAS CNTR IS R-TR OR LCTR.        *   479
%     ORIGINAL MODE WIT IS SAVED IN ORGMD AND SHIFTED ONE IS SAVED IN       *   480
%     SHFMD, AFTER EACH TRANSMISSION OF PARTICLE. THE PARTICLE TABLE        *   481
%                                                                           *   482
%           ENTRY IS LINKED TO VACANCY CHAIN.                               *   483
        BEGIN                                                              *   484
          BOOLEAN       ORGMD,SHFMD;                                       *   485
          CINT          L;                                                 *   486
          PINT          PJ,TMP;                                            *   487
            FOR PJ:=1 STEP 1 UNTIL CNTR DO                                 *   488
              BEGIN                                                        *   489
                ORGMD:=MODE;                                               *   490
                SHFMD:=SHIFTR(K,MODE);                                     *   491
                MODE:=SHFMD;                                               *   492
              LOOP L:=0,1,3 DO   %SEND A PARTICLE TO ITS NEIGHBOR.         *   493
                PKL.[VNXT+L]X1:=RTR(K,TRUE,PKL.[PNTR+L-1]X);               *   494
              VNXT:=PKL.[VNXT]X4;                                          *   495
              VCTR:=VCTR-1;                                                *   496
              MODE:=ORGMD;   %ORIGINAL MODE                                *   497
             PKL.[PNTR+2]X1:=0;  %WEIGHING FACTOR SET TO ZERO              *   498
                            %TO INDICATE THAT A PARTICLE HAS               *   499
                                %BEEN ROUTED.                              *   700
              TMP:=PNTR;    %SWITCH FROM (R OR L)POINTER LINK              *   701
                                %TO VACANCY LINK.                          *   702
              PNTR:=PKL.[PNTR]X4;                                          *   703
              PKL.[TMP-1]X4:=VNXT;                                         *   704
              VNXT:=TMP-1;                                                 *   705
              VCTR:=VCTR+1;   %NEW VACANCY BECAUSE A PARTICLE              *   706
                              %HAS BEEN ROUTED TO ITS NEIGHBOR             *   707
              END;    %ALL PARTICLES  TRANSFERED                           *   708
                                                                  489
              CNTR:=0;                                                     *   709
        END; %% ROUTE %%                                                   *   710
                                                                  484
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   711
        SUBROUTINE TABLEPRNT(CINT II);                                     *   712
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*   713
          BEGIN                                                            *   714
%       TO PRINT OUT THE PARTICLE TABLE                                    *   715
          PREAL     A;                                                     *   716
            SIMWRITE(LINE[PAGE],"PARTICLE TABLE--NO.",II);                 *   717
            SIMWRITE(LINE,"CELL NUMBER",PKL.[II]X3);                       *   718
```

```
                A:=CVTRL(PKL.[II+1]X5);                                          *    719
                SIMWRITE(LINE."DIRECTION COS",A);                                *    720
                SIMWRITE(LINE[PAGE]."POINTER 1",PKL.[II]X4);                     *    721
                SIMWRITE(LINE."POINTER 2",PKL.[II+1]X4);                         *    722
                A:=CVTRL(PKL.[II]X5);                                            *    723
                SIMWRITE(LINE[PAGE]."X-COORDINATE",A);                           *    724
                A:=CVTRL(PKL.[II]X6);                                            *    725
                SIMWRITE(LINE."Y-COORDINATE",A);                                 *    726
                A:=CVTRL(PKL.[II+1]X3);                                          *    727
                SIMWRITE(LINE[PAGE]."SIN PFI",A);                                *    728
                A:=CVTRL(PKL.[II+1]X6);                                          *    729
                SIMWRITE(LINE."COS PFI",A);                                      *    730
                SIMWRITE(LINE[PAGE]."DELTA T",PKL.[II+2]X1);                     *    731
                SIMWRITE(LINE."ENERGY",PKL.[II+2]X2);                            *    732
                SIMWRITE(LINE[PAGE]."SURVIVAL DIST.",PKL.[II+3]X2);              *    733
                SIMWRITE(LINE."WEIGHT",PKL.[II+3]X1);                            *    734
              END;  % PRINT PARTICLE TABLE                                       *    735
                                                        714
              %                                          %                      *    736
%                                                                              *    737
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    738
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    739
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    740
              %                                          %                      *    741
%         BEGINNING OF MAIN PROGRAM                                            *    742
%                                                                              *    743
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    744
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    745
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*    746
          PKL:=GETPER(80);                                                     *    747
          DENS:=GETPFR(8);                                                     *    748
          ENERGY:=GETPER(8);                                                   *    749
          CHECK:=GETPER(1);                                                    *    750
%                                                                              *    751
%                                                                              *    752
          DATALOAD;                                                            *    753
          SUM:=ABC+SCC;                                                        *    754
%                                                                              *    755
%%      COUNTERS CLEARED                                                       *    756
          ACTR:=0;                                                             *    757
          ECTR:=0;                                                             *    758
          VNXT:=0;                                                             *    759
          VCTR:=0;                                                             *    760
          LOOP TNR:=0,4,76 DO       % 20 PARTICLES IN EACH PF.                 *    761
               CHAIN(VCTR,VNXT,0,TNR);  %VACANCY IS LINKED.                    *    762
%                                                                              *    763
%                                                                              *    764
          LOOP TIME:=0,1,3 DO       % BEGINNING OF TIME CYCLE                 *    765
            BEGIN                                                              *    766
               RCTR:=0;                                                        *    767
               LCTR:=0;                                                        *    768
               RNXT:=1;                                                        *    769
               LNXT:=1;                                                        *    770
%                                                                              *    771
%                                                                              *    772
               ENERGYBALANCE;                                                  *    773
               PARTICLEASSIGN;                                                 *    774
```

```
          BIRTH;                                                          *    775
%                                                                         *    776
          LOOP TNB:=0,1,76 DO       %BEGINNING OF TABLE SCANNING          *    777
            BEGIN                                                         *    778
              %%% READ OUT FROM THE PARTICLE  TABLE  %%%                  *    779
              CNB:=PKL.[TNB]X3;          % CELL NUMBER                   *    780
              XCO:=CVTRL(PKL.[TNB]X5)); % X-CO ORDINATE                  *    781
              YCO:=CVTRL(PKL.[TNB]X6)); % Y-COORDINATE                   *    782
              TNB:=TNB+1;                                                 *    783
              MU:=CVTRL(PKL.[TNB]X9)); %DIRECTION COSINE                  *    784
              SNF:=CVTRL(PKL.[TNB]X3)); %SIN PHI                          *    785
              CSF:=CVTRL(PKL.[TNB]X4)); %COS PHI                          *    786
              TNB:=TNB+1;                                                 *    787
              TM:=PKL.[TNB]X1;     %RESIDUAL TIME                         *    788
              ENR:=PKL.[TNB]X2;    %ENERGY                                *    789
              TNB:=TNB+1;                                                 *    790
              WT:=PKL.[TNB]X1;     %WEIGHTING FACTOR                      *    791
              SVD:=PKL.[TNB]X2;    %SURVIVAL DISTANCE                     *    792
              DPTH:=MOD(CNB);                                             *    793
              FOR ALL AT GF, .01 DO                                       *    794
                   CHECK.A5:=8192;      %ENABLE PE'S IN BRANCH            *    795
              BRANCH(TNB);                                                *    796
%                                                                         *    797
              STORE(TNB);                                                 *    798
              CHECK.X7:=0;                                                *    799
            END;     % TABLE SCANNING                                    *    800
                                                                  772
          SCATTER;                                                        *    801
          ROUTE(RCTR,RNXT,1);                                             *    802
          ROUTE(LCTR,LNXT,0));                                            *    803
          TALLY;                                                          *    804
        END;  %TIME CYCLE                                                 *    805
                                                                  764
    END. % OF MAIN                                                        *    806
                                                                    1
```

```
$**********.**********APPROXIMATE PROGRAM SIZE**********.**********

$                       INSTRUCTION SYLLABLES GENERATED.....5383
$                       BOOLEAN STORAGE.....10 WORDS PER QUADRANT
$                       CU REGISTER STORAGE.....25 WORDS
$                       PE REGISTER STORAGE.....6464 WORDS PER QUADRANT
$                       MISCELLANEOUS.....3680 WORDS
$                       ***** TOTAL.....13766 WORDS


CONGRATULATIONS. NO SOURCE PROGRAM ERRORS WERE DETECTED.

          TOTAL TIME..... 1:12
          CPU TIME..... 0:39
          I/O TIME..... 0:28
          806 CARDS PROCESSED AT 1211 C.P.M. (CPU)
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author) Center for Advanced Computation University of Illinois at Urbana-Champaign Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

REPORT TITLE

PARALLEL RADIATION TRANSPORT CODE
A MONTE CARLO METHOD APPLIED TO ILLIAC IV AND ITS STATISTICAL CONSIDERATION

DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research Report

AUTHOR(S) (First name, middle initial, last name)

Ken'ichi Miura

REPORT DATE October 1, 1971	7a. TOTAL NO. OF PAGES 95	7b. NO. OF REFS 17
8a. CONTRACT OR GRANT NO. DAHC04 72-C-0001 b. PROJECT NO. ARPA Order 1899 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) CAC Document No. 18 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT

Copies may be obtained from the address given in (1) above.

11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY U.S. Army Research Office-Durham Duke Station Durham, North Carolina

13. ABSTRACT

A Monte Carlo method to solve radiation transport problems by using the ILLIAC IV is discussed. An emphasis is put on the special features to be encountered in parallel computation: data structure, PE efficiency, etc. A test program implemented in GLYPNIR--an ILLIAC IV language--is shown and some preliminary results on its statistical properties are also discussed.

DD FORM 1 NOV 68 1473

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Physics; Nuclear Sciences						
Procedure and Problem Oriented Languages (GLYPNIR)						
Ordinary and Partial Differential Equations						